FLAG: Flow Policy MaxEnt-RL by Latent Augmented Guidance

Sungha Kim[1*] **Gawon Lee**[2*] **Jusuk Lee**[2]
Jonghae Park[2] **H. Jin Kim**[1, 2] **Daesol Cho**[3†]
[1] Interdisciplinary Program in Artificial Intelligence (IPAI), Seoul National University
[2] Department of Aerospace Engineering (AE), Seoul National University
[3]Georgia Institute of Technology
{rlatjdgk0307, lgw1997}@snu.ac.kr

Abstract

Maximum entropy reinforcement learning (MaxEnt-RL) enables robust exploration, yet practical implementations often restrict policies to simple Gaussians. While recent approaches incorporate expressive generative policies via importance-weighted supervised learning, they are prone to importance weight collapse, which limits their scalability in high-dimensional action spaces. Our key insight is to mitigate this limitation by localizing the sampling region, avoiding the weight degeneracy induced by importance sampling over the entire action space. To instantiate this insight, we introduce **FLAG** (**F**low policy with **L**atent-**A**ugmented **G**uidance). FLAG augments the state space with a flow latent variable and optimizes a provably consistent proxy MaxEnt-RL objective. We empirically demonstrate that FLAG enables expressive policy optimization with limited importance samples and scales to high-dimensional control tasks. Furthermore, FLAG achieves state-of-the-art performance across challenging benchmarks. Our project webpage: https://flag-rl.github.io/

1 Introduction

Maximum entropy reinforcement learning (MaxEnt-RL) enhances conventional reward maximization with an entropy regularization term, enabling robust decision-making and persistent exploration [50, 45, 19, 20]. By encouraging stochasticity in the policy, MaxEnt-RL can represent multiple near-optimal behaviors through expressive action distributions [19]. Most existing methods parameterize the policy as a Gaussian distribution due to its optimization efficiency [20, 7, 32]. However, the unimodal nature of Gaussian policies fundamentally limits their ability to capture the complex optimal policies induced by the MaxEnt-RL objective. To address this expressivity bottleneck, recent studies have introduced diffusion- and flow-based generative policies, achieving strong performance on high-dimensional continuous control benchmarks [47, 11].

Despite their empirical success, these approaches typically require Backpropagation Through Time (BPTT) across multiple generative steps, which can suffer from numerical instability [6, 34]. To avoid these issues, a separate line of work [16, 27, 12] adopts an Expectation-Maximization (EM, [29]) framework in the vein of MPO [2], casting policy optimization as supervised learning on the target distribution. While this strategy circumvents BPTT, it still relies on importance sampling (IS, [41]) because sampling from the MaxEnt target distribution is intractable. In high-dimensional spaces, the proposal–target mismatch can cause importance weights to explode, exacerbating the vanishing support problem [30, 42]. Prior works [16, 27, 12, 13] heuristically constrain IS weights for stability,

*Equal contribution.
†Corresponding Author.

Preprint.

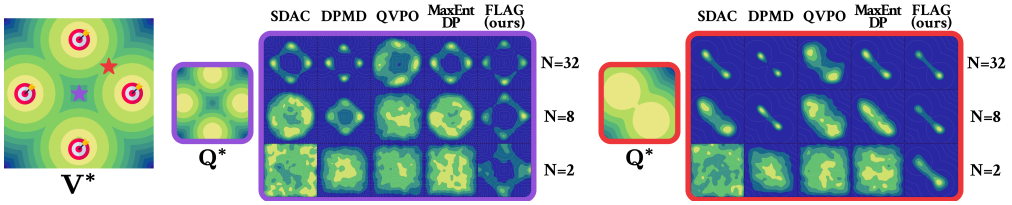


Figure 1: **Comparisons with global policy sampling methods in multi-goal environment.** The multi-goal environment [19] tasks a point mass with navigating to one of four symmetrically placed targets. We plot the optimal value function $\mathbf{V}^*$ and Q-functions $\mathbf{Q}^*$ at two different states (★ & ★), along with policies at each state. While other methods fail to learn with fewer samples ($N \leq 8$), **FLAG** captures optimal and multi-modal behaviors leveraging latent-augmented guidance, even in limited importance sample budget commonly arising in high-dimensional action spaces. Additional details on the environment and setup are presented in Appendix G.1.

but these post-hoc corrections do not directly increase the probability of sampling target-relevant actions. As a result, IS over the entire action space, which we refer to as *global IS*, remains prone to weight degeneracy, leading to poor sample efficiency and sparse supervision insufficient for tracking non-stationary targets in online RL. We illustrate this sampling inefficiency with a didactic multigoal example under limited sample budgets N (Figure 1), where global IS baselines fail to capture the target mode as N decreases from 32 to 2. We observe the same trend in high-dimensional control (Figure 3).

Our key insight is to mitigate the sparsity of global IS by jointly localizing the proposal and target distributions. By conditioning on a latent variable, both distributions are restricted to the same local region, allowing IS to operate locally rather than over the full action space, which we refer to *local IS*. To achieve this, we leverage the deterministic nature of flow-matching models, which induces a fixed mapping between latent vectors and actions. Based on this property, we construct a Gaussian *local policy* centered on the flow's output, which admits a formal definition of a latent-augmented MDP where the standard state space is extended to include the latent variable. Within this framework, the *global policy* is recovered by marginalizing over the latent space, providing a theoretical foundation for optimization via the local IS rather than the global IS.

To instantiate this insight, we propose **FLAG** (**F**low policy MaxEnt-RL by **L**atent **A**ugmented **G**uidance), which trains the flow policy in a supervised manner on the latent-augmented MDP. Specifically, we introduce a cross-entropy-based proxy MaxEnt objective that avoids the intractable entropy computation of the global policy, and show that it remains consistent with the original MaxEnt objective when the local policy variance is properly controlled during training. We then optimize the proxy objective with an EM-style procedure, using the resulting locally updated policies as latent-conditioned supervised targets for the flow policy, which we call *latent-augmented guidance*. This latent-conditioned local IS allows FLAG to capture the target mode with only $N = 2$ samples in Figure 1, suggesting better scalability to high-dimensional action spaces (Figure 3).

In summary, our contributions are threefold:

- We introduce the latent-augmented MDP to derive local IS and derive a proxy MaxEnt objective on this augmented MDP, with a theoretical consistency with respect to the original objective.
- We propose FLAG, which optimizes the proxy MaxEnt objective via an EM algorithm where samples are drawn by local IS, and prove that FLAG monotonically improves the proxy objective.
- We empirically demonstrate that FLAG outperforms global IS baselines and achieves state-of-the-art performance across challenging benchmarks.

2 Related Works

Action Gradient. DIPO [49] and QSM [38] leverage Q-gradients ($\nabla_a Q$) to construct supervised policy updates, either by refining replay-buffer actions or matching policy scores. However, their reliance on Gaussian noise exploration motivates a more principled MaxEnt-RL framework.

BPTT-based Actor-Critic. Recent methods treat generative models as parameterized policies within actor-critic frameworks. For example, DACER [47, 48] optimizes Q-values along reverse diffusion, while DIME [11] optimizes Q-values along with a tractable marginal-entropy lower

bound. However, these objectives require BPTT through generation, making optimization memory-intensive and numerically unstable. FlowRL [26] avoids BPTT using a single-step policy with implicit optimality guidance, but its objective lacks exploration and can be suboptimal. In contrast, our method derives latent-augmented guidance and optimizes the proxy MaxEnt-RL objective with supervised updates, bypassing BPTT without sacrificing the expressivity of the underlying generative model.

Target Policy Matching. Supervised policy updates in RL have been widely studied under the probabilistic inference framework [24], including EM-based methods such as RWR [37], AWR [36], and MPO [2]. Recent generative-policy extensions, such as RSM [27] and MaxEntDP [16], use reweighted score matching or Q-weighted noise estimation to project policies toward target distributions. However, they rely on global IS, which disperses samples over the full action space and provides sparse supervision in regions important for policy improvement. FLAG instead uses local IS to concentrate samples around each action, yielding denser and more effective policy supervision.

Policy Gradient. FPO [28] and GenPO [14] adapt PPO [40] to generative policies, but are on-policy methods. Our closest off-policy baseline is QVPO [13], which optimizes a weighted diffusion loss to improve the policy. QVPO's weighted score-matching update relies on a global proposal distribution, inheriting the sparsity issue of global IS, where target-relevant actions are rarely sampled; in contrast, FLAG uses a localized proposal for policy improvement.

3 Preliminaries

3.1 Reinforcement Learning as Probabilistic Inference

Notation. We consider a Markov Decision Process (MDP) defined by the tuple $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote the continuous state and action spaces, $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ represents the transition probability density, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. We define a stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$, mapping states to probability densities over actions. In the infinite-horizon setting, π induces $p_\pi(\tau)$ and discounted marginals $\rho_\pi(s)$ and $\rho_\pi(s, a)$.

Control-as-inference and MaxEnt-RL. RL admits a probabilistic inference view via binary optimality variables O_t, where $O_t = 1$ indicates (s_t, a_t) is optimal [24]. With $\mathcal{O} = \{O_t = 1\}_{t=0}^{\infty}$ and $p(\mathcal{O} \mid \tau) \propto \exp(\sum_t \gamma^t r(s_t, a_t)/\alpha)$ for temperature $\alpha > 0$, Bayes' rule gives the optimal posterior

$$p^\star(\tau \mid \mathcal{O}) \propto p(s_0) \prod_{t=0}^{\infty} p(a_t \,|\, s_t) p(s_{t+1} \,|\, s_t, a_t) \exp\left(\sum_{t=0}^{\infty} \gamma^t \big(r(s_t, a_t)/\alpha \big) \right). \tag{1}$$

Matching $p_\pi(\tau)$ to this posterior via KL-divergence recovers the standard MaxEnt-RL objective [20]:

$$\pi^\star = \arg\max_\pi \mathbb{E}_{\tau \sim p_\pi} \left[\sum_{t=0}^{\infty} \gamma^t \big(r(s_t, a_t) - \alpha \log \pi(a_t \,|\, s_t) \big) \right]. \tag{2}$$

EM-style Policy Optimization. Introducing a variational policy $q(a \mid s)$ that shares the initial-state distribution and dynamics with p_π, the trajectory-level KL reduces to action-only terms, yielding the variational lower bound [1, 2]

$$\mathcal{J}(q, \pi) = \mathbb{E}_{\tau \sim p_q} \left[\sum_{t=0}^{\infty} \gamma^t \big(r(s_t, a_t)/\alpha \big) \right] - D_{\mathrm{KL}}(p_q(\tau) \,\|\, p_\pi(\tau)). \tag{3}$$

For the current parameterized policy $\pi_k = \pi_{\theta_k}$, the KL-constrained E-step

$$q_k(\cdot \,|\, s) = \arg\max_{q \in \Delta(\mathcal{A})} \mathbb{E}_{a \sim q} \left[Q^{\pi_k}(s, a) \right] \qquad \text{s.t. } D_{\mathrm{KL}} \left(q(\cdot \,|\, s) \,\|\, \pi_k(\cdot \,|\, s) \right) \le \epsilon \tag{4}$$

admits a closed form $q_k(a \,|\, s) \propto \pi_k(a \,|\, s) \exp\left(Q^{\pi_k}(s, a)/\lambda_k \right)$, with $\lambda_k > 0$ the dual variable. The M-step projects q_k back into the parametric class, yielding a weighted maximum-likelihood:

$$\theta_{k+1} \in \arg\max_\theta \mathbb{E}_{s \sim \rho_{\pi_k}} \mathbb{E}_{a \sim q_k(\cdot|s)} \left[\log \pi_\theta(a \mid s) \right], \tag{5}$$

typically estimated via importance weighting on samples from π_k.

3.2 Conditional Flow Matching

Flow matching [25] transports a prior p_0 to a target p_1. It learns a time-conditioned vector field $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. The flow ψ is governed by the following ordinary differential equation (ODE):

$$\frac{d}{dt}\psi^t(x) = v^t(\psi^t(x)) \quad \textit{i.e.} \quad \psi^t(x) = \psi^0(x) + \int_0^t v^\tau(\psi^\tau(x))d\tau. \tag{6}$$

By parameterizing the vector field v with a parameter θ and under the Optimal Transport linear path, the vector field can be directly optimized via the Conditional Flow Matching (CFM) objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_0,x_1}\Big[\big\|v_\theta^t\big((1-t)x_0 + tx_1\big) - (x_1 - x_0)\big\|^2\Big], \tag{7}$$

where $t \sim \mathcal{U}[0,1]$, $x_0 \sim p_0$, and $x_1 \sim p_1$.

4 Method

FLAG targets the optimization of a flow policy guided by local supervision from a Gaussian local policy. First, we introduce a policy parameterization (4.1) that explicitly couples the flow-based anchor actions with a local Gaussian distribution. Next, we lift the original environment to a latent-augmented MDP (4.2). This augmented construction makes the local conditioning variable explicit and establishes that optimizing the local policy in the lifted MDP is mathematically equivalent to optimizing the global MaxEnt policy. On top of this equivalence, we derive an EM-based policy improvement procedure (4.3) that admits the latent-augmented guidance. Finally, we introduce a practical algorithm (4.4) that enables updates of the flow policy via the latent-augmented guidance. All derivations in this section are in Appendix B.

4.1 Policy Parameterization

Our policy parameterization centers the Gaussian on the flow's anchor $T_\theta(s,z)$, which localizes the effective sampling region around each latent z, providing dense supervision for EM updates and circumventes the importance-weight degeneracy of global proposal sampling. We first define a flow transformation $T_\theta(s,z)$ parameterized by θ that maps a latent vector $z = a^0 \sim p_z$ to an anchor action $T_\theta(s,z) = a^1 = z + \int_0^1 v_\theta(a^\tau, \tau, s)\, d\tau$ following Eq. (6), inducing the base flow policy $\tilde{\pi}_\theta(a \mid s) = p_z(z)\,|\det(\partial T_\theta(s,z)/\partial z)|^{-1}$.

Next, we augment each flow-generated anchor action $T_\theta(s,z)$ with an auxiliary local Gaussian:

$$\hat{\pi}(a \mid s, z; \theta) = \mathcal{N}(a;\, T_\theta(s,z),\, \Sigma). \tag{8}$$

For theoretical development in this paper, we treat Σ as non-learnable, isotropic covariance whose schedule is specified in Section 4.4; a learnable variant is discussed in Appendix D.4. Marginalizing over the prior recovers the global policy:

$$\pi(a \mid s;\, \theta) = \int p_z(z)\, \hat{\pi}(a \mid s, z;\, \theta)\, dz. \tag{9}$$

4.2 Latent augmented MDP and Proxy MaxEnt-RL Objective

In this section, we show that local policy optimization on the augmented MDP is mathematically equivalent to maximizing the original MaxEnt-RL objective in the original MDP. We start by defining the latent-augmented MDP.

Definition 4.1 (Latent-augmented MDP)**.** The latent augmented MDP $\hat{\mathcal{M}}$ is defined by the state $\hat{s} = (s,z) \in \mathcal{S} \times \mathcal{Z}$ and action $a \in \mathcal{A}$. Since z is sampled from a prior p_z which is independent of state, the transition dynamics $\hat{p}$ and reward function $\hat{r}$ are given as:

$$\hat{p}\,(\hat{s}' \mid \hat{s}, a) = p(s' \mid s, a)\, p_z(z'), \qquad \hat{r}(\hat{s}, a) = r(s,a). \tag{10}$$

We hereafter refer to this construction as the **z-MDP**, and adopt the notation $(\hat{\cdot})$ to distinguish entities in the z-MDP from those in the original MDP. In Corollary 4.2, we show that any expectation over a measurable function is unchanged whether you average over $\hat{\rho}_{\hat{\pi}}(\hat{s}, z)$ or $\rho_\pi(s,z)$.

Corollary 4.2 (Marginal consistency). *Given $\hat{\pi}$ and π in Section 4.1, the discounted state–action marginal distribution $\hat{\rho}_{\hat{\pi}}(s,z,a)$ in the z-MDP $\hat{\mathcal{M}}$ satisfies $\rho_\pi(s,a) = \int \hat{\rho}_{\hat{\pi}}(s,z,a)dz$. Consequently, the following equation holds for any measurable function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$*

$$\mathbb{E}_{(s,z,a)\sim\hat{\rho}_{\hat{\pi}}}\left[f(s,a)\right] = \mathbb{E}_{(s,a)\sim\rho_\pi}\left[f(s,a)\right]. \tag{11}$$

Cross-entropy. The MaxEnt-RL objective requires the intractable entropy of π, as the marginalization over z in Eq. (9) has no closed form. To circumvent this, we propose using the cross-entropy between π and $\tilde{\pi}_\theta$, $\mathcal{H}\left(\pi(\cdot\,|\,s),\, \tilde{\pi}_\theta(\cdot\,|\,s)\right) \triangleq \mathbb{E}_{a\sim\pi(\cdot|s)}\left[-\log\tilde{\pi}_\theta(a \mid s)\right]$, which can be evaluated in an unbiased manner via the Hutchinson trace estimator (see Appendix A for details). The cross-entropy is reformulated as an expectation over z:

$$\mathcal{H}\left(\pi(\cdot\,|\,s), \tilde{\pi}_\theta(\cdot\,|\,s)\right) = \mathbb{E}_{z\sim p_z}\left[\mathcal{H}\left(\hat{\pi}(\cdot\,|\,\hat{s}),\, \tilde{\pi}_\theta(\cdot\,|\,s)\right)\right] \tag{12}$$

For brevity, we denote $\tilde{\pi}_\theta$ by $\tilde{\pi}$ throughout this section. The following proposition establishes that the cross-entropy converges to the entropy of π.

Proposition 4.3 (Cross-Entropy as a Valid Entropy Surrogate). *For smooth $\tilde{\pi}$ and $\mathrm{tr}\left(\Sigma\right) \ll 1$, the local variance governs the Total Variation (TV) distance and KL divergence between π and $\tilde{\pi}$:*

$$D_{\mathrm{TV}}(\pi,\tilde{\pi}) = \mathcal{O}\big(\sqrt{\mathrm{tr}(\Sigma)}\big), \qquad D_{\mathrm{KL}}(\pi \parallel \tilde{\pi}) = \mathcal{O}\left(\mathrm{tr}(\Sigma)^2\right). \tag{13}$$

Consequently, as the local variance vanishes ($\mathrm{tr}(\Sigma) \to 0$), the surrogate cross-entropy converges to the true entropy of the global policy:

$$\mathcal{H}(\pi,\tilde{\pi}) = \mathcal{H}(\pi) + \mathcal{O}\left(\mathrm{tr}(\Sigma)^2\right) \approx \mathcal{H}(\pi). \tag{14}$$

Proposition 4.3 justifies a tractable proxy for the MaxEnt-RL objective in $\mathcal{M}$ via the cross-entropy under a small-variance assumption on the local Gaussian.

Proxy MaxEnt-RL Objective. Using the cross-entropy, we now define the proxy MaxEnt-RL objective and corresponding soft value functions. The augmented policy objective is given as

$$\mathcal{J}(\pi) = \sum\nolimits_{t=0}^{\infty} \gamma^t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\big(r(s_t,a_t) - \alpha\log\tilde{\pi}(a_t \mid s_t)\big)\right]. \tag{15}$$

By Corollary 4.2 and Eq. (12), this objective is equivalently evaluated in the z-MDP as $\mathcal{J}(\hat{\pi})$, providing a theoretical bridge for optimization within the latent-augmented space $\hat{\mathcal{M}}$.

Value function. Following SAC [20], we define the soft state-value function and the corresponding soft Bellman operator $\mathcal{T}^\pi$ by replacing the entropy term with the cross-entropy term:

$$\text{(Soft Value Function)} \qquad V^\pi(s) = \mathbb{E}_{a\sim\pi}\left[Q^\pi(s,a) - \alpha\log\tilde{\pi}(a \mid s)\right] \tag{16}$$

$$\text{(Soft Bellman Operator)} \quad (\mathcal{T}^\pi Q)(s,a) \triangleq r(s,a) + \gamma\mathbb{E}_{s'\sim p}\left[V^\pi(s')\right] \tag{17}$$

In the z-MDP $\hat{\mathcal{M}}$, the Q-function is defined with $Q^{\hat{\pi}}\left(\hat{s},a\right)$.

Corollary 4.4 (Q-function consistency). *By Corollary 4.2, we use the same Q-function both in $\mathcal{M}$ and $\hat{\mathcal{M}}$*

$$Q^{\hat{\pi}}(\hat{s},a) = Q^\pi(s,a), \quad \forall s,z,a \tag{18}$$

Therefore, we use the Q-function $Q^\pi(s,a)$ of the original MDP in z-MDP.

According to Corollaries 4.2 and 4.4, maximizing the Q-function $Q^\pi(s,a)$ via the local policy $\hat{\pi}$ for all $\hat{s} \sim \hat{\rho}_{\hat{\pi}}$ in the z-MDP is equivalent to optimizing the global policy π for all $s \sim \rho_\pi$.

4.3 FLAG: A Practical Local Policy Update Algorithm

In this section, we extend the EM updates from Section 3.1 to the z-MDP and our local policy $\hat{\pi}(\cdot \mid \hat{s};\theta)$. At each iteration k, we treat $\tilde{\pi}_{\theta_k}$ as a fixed reference, making the augmented reward $r_t - \alpha\log\tilde{\pi}_{\theta_k}(a_t \mid s_t)$ play the role of an iteration-fixed reward, enabling the EM derivation

$$\mathcal{J}_{\mathrm{EM}}(q,\theta) = \mathbb{E}_{\hat{\tau}\sim p_q}\left[\sum\nolimits_{t=0}^{\infty}\gamma^t\big((r_t - \alpha\log\tilde{\pi}_\theta(a_t\,|\,s_t))/\lambda\big)\right] - D_{\mathrm{KL}}\big(p_q(\hat{\tau}) \,\|\, p_{\hat{\pi}}(\hat{\tau})\big). \tag{19}$$

Crucially, the cross-entropy structure allows us to define and evaluate this objective for a variational distribution q, which admits E-step target in a closed form. For a reference policy $\hat{\pi}_k$ parameterized by θ_k and its associated Q-function Q^{π_k}, we define the energy function for brevity

$$f_{\hat{s},k}(a) := Q^{\pi_k}(s,a) - \alpha \log \tilde{\pi}_{\theta_k}(a \mid s). \tag{20}$$

For each $\hat{s} \sim \hat{\rho}_{\hat{\pi}_k}$,

$$\text{(E-step:)} \qquad q_k(a \mid \hat{s}) \propto \hat{\pi}(a \mid \hat{s}; \theta_k) \exp(f_{\hat{s},k}(a)/\lambda) \tag{21}$$

$$\text{(M-step:)} \qquad \theta_{k+1} \in \arg\max_{\theta} \mathbb{E}_{a \sim q_k}\left[\log \hat{\pi}(a \mid \hat{s}; \theta)\right] \tag{22}$$

M-step via Moment Matching. Since $\hat{\pi}$ is Gaussian with fixed Σ_k, the M-step in Eq. (22) reduces to matching the first moment :

$$\theta_{k+1} \in \arg\min_{\theta} \mathbb{E}_{(s,z)\sim\hat{\rho}_{\hat{\pi}}} \|T_\theta(s,z) - \mu_k^*(\hat{s})\|^2, \quad \mu_k^*(\hat{s}) := \mathbb{E}_{q_k}[a]. \tag{23}$$

We approximate μ_k^* via self-normalized importance sampling [8] with N samples $\{\delta_i\}_{i=1}^N \sim \hat{\pi}_k(\cdot \mid \hat{s})$:

$$\mu_k^*(\hat{s}) = \mathbb{E}_{q_k}[a] \approx \sum\nolimits_{i=1}^{N} \bar{w}_i a_i, \quad \bar{w}_i = \mathrm{softmax}\left(f_{\hat{s},k}(a+\delta_i)/\lambda\right). \tag{24}$$

We treat $\mu_k^*(\hat{s})$ as an improved action label for the anchor action $T_{\theta_k}(s,z)$. To realize Eq. (23), we distill these labels into the base flow policy $\tilde{\pi}_{\theta_k}$ using the CFM objective in Eq. (7), i.e.,

$$\mathcal{L}(\theta; s, z) = \mathbb{E}_\tau\left[\|u_\theta^\tau(s, \zeta^\tau(z, \mu^*)) - (\mu^* - z)\|^2\right]. \tag{25}$$

This label acts as a *guiding flag*, as it indicates the latent vector z where to generate in the action space (Figure 2).

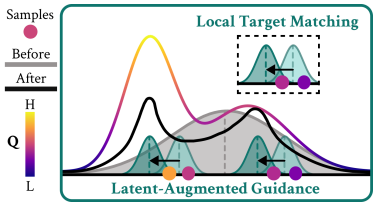


Figure 2: Target matching is performed on the local policy and target actions are distilled back to the flow policy.

Monotonic improvement guarantee. We now show that FLAG's EM update monotonically improves the variational objective, $\mathcal{J}_{k+1} \geq \mathcal{J}_k$, where $\mathcal{J}_k := \mathcal{J}_{\mathrm{EM}}(q_k, \theta_k)$. Let $\mathcal{G}_k$ denote the squared gradient norm of the Eq. (23) and the local covariance be $\Sigma_k = \sigma_k^2 I$.

Theorem 4.5 (Monotonic Improvement Guarantee)**.** *Let the CFM update (Eq. (25)) approximates the ideal KL projection (Eq. (23)) up to an error $\epsilon_k^{\mathrm{proj}}$. For sufficiently small step size β the one-step improvement of the FLAG update is lower-bounded by*

$$\mathcal{J}_{k+1} - \mathcal{J}_k \;\geq\; \underbrace{\lambda\beta\mathcal{G}_k}_{\substack{\text{frozen-reward} \\ \text{MPO improvement}}} \;-\; \underbrace{\alpha\, C_\Sigma \sigma_k^2\, \beta\mathcal{G}_k}_{\substack{\text{cross-entropy} \\ \text{reward drift}}} \;-\; \underbrace{\lambda\, \epsilon_k^{\mathrm{proj}}}_{\substack{\text{CFM projection} \\ \text{error}}} \;+\; \mathcal{O}(\beta^2), \tag{26}$$

where $C_\Sigma > 0$ is a constant absorbing zeroth-order approximation residuals. Consequently, whenever $\lambda > \alpha C_\Sigma \sigma_k^2$ and $\epsilon_k^{\mathrm{proj}}$ is sufficiently small, $\mathcal{J}_{k+1} \geq \mathcal{J}_k$.

The three terms in Theorem 4.5 correspond to the standard MPO improvement under frozen reward, the drift of the cross-entropy reward as θ_k changes, and the approximation error introduced by the CFM realization of the KL projection (see Figure 6 for the proof overview). We ensure the sufficient conditions $\lambda > \alpha C_\Sigma \sigma_k^2$ and $\epsilon_k^{\mathrm{proj}} \to 0$ are satisfied through the design choices in Section 4.4. While Theorem 4.5 follows variational framework, Figure 5 and Appendix C.2 provide a complementary derivation showing how FLAG's moment-matching update recovers SAC's soft policy improvement.

4.4 Implementation Details

Autotuning Temperature. We follow [20] to automatically tune the entropy scaling term α:

$$\mathcal{L}(\alpha) = \alpha(\mathcal{H}(\pi, \tilde{\pi}) - \mathcal{H}_{\text{target}}), \tag{27}$$

where $\mathcal{H}_{\text{target}}$ is a hyperparameter that determines the stochasticity of the policy.

Effective Temperature. We normalize the energy function as $f_{\mathrm{norm}}(a) = f_{\hat{s},k}(a)/\alpha$ with a fixed reference temperature λ_{ref} [17, 22, 26], equivalent to using an effective temperature $\lambda = \alpha\lambda_{\mathrm{ref}}$ in

Eq. (21). Fixing λ_{ref} avoids the numerical instability of optimizing two Lagrange multipliers α and λ, and reduces the sufficient condition for monotonic improvement (Theorem 4.5) to $\lambda_{\text{ref}} > C_\Sigma \sigma_k^2$.

Covariance Scheduling. We use an isotropic covariance $\Sigma_k = \sigma_k^2 I$. σ_k is annealed from σ_{init} to a small σ_{final}, keeping σ_k small throughout training. This scheduling is crucial in Theorem 4.5: since the reward-drift term scales as $\mathcal{O}(\sigma_k^2)$, annealing $\sigma_k \to 0$ tightens the guarantee.

Guidance Buffer. We cache locally improved action labels $\mu_k^*(\hat{s})$ in a small guidance buffer and reuse them across multiple off-policy updates [39, 44]. Since the E-step targets drift slowly across iterations, recent labels remain effective targets for the current policy update, providing dense supervision that keeps the CFM projection error ϵ_k^{proj} in Theorem 4.5 small in practice.

Other Details. Following DIME [11], we employ a distributional critic [5] trained via CrossQ [7]. Full implementation details and the training procedure are provided in Appendix D and Algorithm 1.

FLAG realizes local IS through EM on the z-MDP.

The z-MDP augments the original MDP with a latent variable while preserving marginal and Q-function consistency. On this augmented MDP, we define a tractable proxy MaxEnt-RL objective whose EM-style optimization induces a localized proposal-target pair conditioned on z. FLAG instantiates this procedure, enabling the local IS within the localized region of the action space and making MaxEnt-RL via supervised learning scalable.

5 Experimental Results

We answer three questions about FLAG in this section.

(Q1) Section 5.1: Does FLAG scale to high-dimensional action spaces under limited sample budgets?

(Q2) Section 5.2: How does FLAG compare to action-gradient and BPTT-based actor-critic methods?

(Q3) Section 5.3: How do our key design choices—covariance scheduling and the guidance buffer—connect to the theoretical results?

We defer additional ablation studies and analyses to Appendix F, which includes: (i) variance reduction of the Hutchinson trace estimator (F.1), (ii) the effect of cross-entropy (F.2), (iii) variations on λ_{ref} (F.3), (iv) the number of ODE steps (F.4), (v) learned covariance and zeroth-order gradient variants (F.5), and (vi) GPU memory consumption (F.6). Hyperparameters and experiment details are deferred to Appendix G.

Benchmarks and Metrics. We evaluate online RL algorithms on three challenging benchmark suites: DMC suite [46], MyoSuite [10], and MuJoCo [43]. MuJoCo is included because target-matching policy optimization methods are commonly evaluated in this domain. We evaluate on four tasks each: HalfCheetah, Walker2d, Ant, and Humanoid for MuJoCo-v5; Dog-Stand, Dog-Walk, Dog-Trot, and Dog-Run for DMC Dog; and Reach-Hard, Obj-Hold-Hard, Key-Turn-Hard, and Pen-Twirl-Hard for MyoSuite. To compare results across benchmarks with different reward scales, we report normalized scores: MuJoCo scores are normalized by CrossQ performance [7], DMC Dog returns are divided by the maximum return of 1,000, and MyoSuite scores are reported as success rates.

5.1 FLAG is Scalable to High-dimensional Action Spaces and Robust to Sample Budget

Is FLAG scalable to high-dimensional action space?

To answer the question, we first compare FLAG with the global IS baselines: SDAC and DPMD [27], QVPO [13], and MaxEntDP [16]. This comparison highlights the limited scalability of the global IS in high-dimensional action spaces under the same conditions. QVPO is included as a baseline since it also obtains samples from global proposal distribution to estimate a lower bound on the policy gradient. All methods use the same CrossQ [7] critic with distributional critics [5], and the number of Q-function evaluations per policy update (N) is fixed to 8 due to computational cost. Our main comparison sets the best-of-P budget to P$=1$, where best-of-P selects the highest-Q action among P

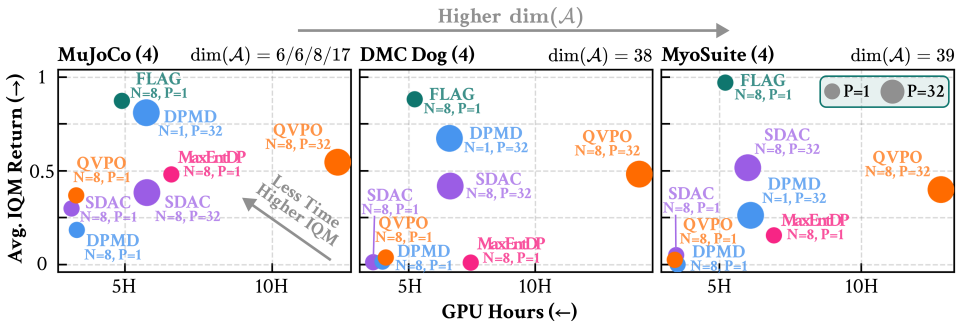


Figure 3: **Performance and computational efficiency of global and local proposal sampling.** We plot normalized performance against wall-clock GPU runtime to examine the performance–efficiency trade-off. The x-axis denotes the time (hours) for a single 1M-step training run on an NVIDIA L40S GPU, and the y-axis reports aggregated IQM returns across 10 random seeds at 1M steps. A method in the *upper-left* region is both higher-performing and takes less time to train. P denotes the number of samples in the best-of-P heuristic, and is indicated by the diameter of each marker. FLAG consistently occupies the upper-left region across all domains, demonstrating superior performance *without* additional computational overhead.

sampled candidates, to isolate the effect of the proposal distribution. We additionally report P = 32 variants where applicable.[3]

As shown in Figure 3, global-proposal baselines struggle in high-dimensional action spaces, especially without best-of-P. Even with the stronger P = 32 heuristic, their performance remains substantially below FLAG. In contrast, FLAG achieves consistently higher returns with the same Q-function evaluation budget (N = 8), indicating that local proposal matching is the key mechanism enabling scalable target matching in high-dimensional control. We also present the learning curves in Figure 10 and Figure 11.

To rule out the possibility that FLAG's advantage comes from the use of CrossQ or from an artificially restricted Q-function inference budget, we evaluate methods without CrossQ. In this setting, the baselines use their default, larger sample budgets. As shown in Figure 4, the global-proposal baselines still fail to learn meaningful behaviors on challenging DMC Dog tasks, whereas FLAG remains effective. This suggests that the performance gap is not explained by the value-learning architecture or the evaluation budget, but by the proposed local proposal matching strategy.

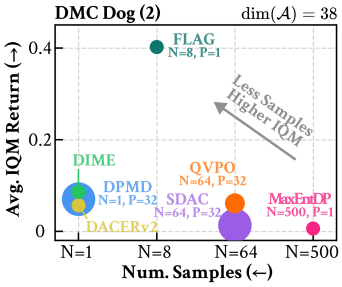


Figure 4: Comparison on the DMC Dog-run and Dog-trot task w/o CrossQ, performance averaged. We report with 10 random seeds for P = 1 and 5 for P = 32. We defer the experimental details to G.2.1.

Is FLAG robust to the sample budget?

We evaluate the sensitivity of FLAG to the sample budget. As shown in Table 1, FLAG remains robust even with limited samples (N = 2), exhibiting only mild performance degradation under limited sampling budgets. This robustness provides empirical evidence for our main design principle: by localizing both the proposal and target distributions, FLAG reduces their discrepancy and makes importance sampling effective within a restricted region of the action space. In contrast to global IS, whose samples are spread over the full action space and may have little overlap with the target density, local IS operates between two distributions supported on the same local region. As a result, the importance weights remain informative even with few samples.

Table 1: Ablation study on the number of training samples in DMC Dog tasks. We aggregate 10 random seeds at 1M steps. The highest and second scores are highlighted.

Num. Samples	Dog-trot Return (1k)	Dog-run Return (1k)
N = 2	0.620 ± 0.072	0.818 ± 0.095
N = 4	0.639 ± 0.061	0.827 ± 0.106
N = 8	0.680 ± 0.066	0.912 ± 0.047
N = 16	0.687 ± 0.073	0.874 ± 0.087
N = 32	0.641 ± 0.075	0.879 ± 0.057

[3]The P = 32 variant is not applied to MaxEntDP because its policy-update candidates are drawn from the replay buffer, making best-of-P selection inapplicable.

Table 2: IQM final performance after 1M environment steps on DMC Dog and MyoSuite. All results are computed from evaluation collected over 10 seeds with 5 evaluation episodes. Each entry reports the IQM point estimate with a 95% confidence interval; the ± notation is used only as a compact summary of the confidence interval ([23]). The highest and second IQM point estimates are highlighted in each column, respectively.

			DMC Dog (4)				
Policy	**Method**	**Alg.**	**Dog-stand**	**Dog-walk**	**Dog-trot**	**Dog-run**	**Avg.**
Gauss.	-	CrossQ	0.957 ± 0.121	0.882 ± 0.079	0.890 ± 0.016	0.467 ± 0.080	0.799
Express.	Action Gradient	DIPO	0.906 ± 0.030	0.570 ± 0.164	0.404 ± 0.080	0.264 ± 0.017	0.536
		QSM	0.061 ± 0.131	0.137 ± 0.144	0.142 ± 0.053	0.064 ± 0.055	0.101
	BPTT-based Actor-Critic	DACERv2	0.923 ± 0.031	0.726 ± 0.258	0.367 ± 0.226	0.090 ± 0.108	0.526
		DIME	0.968 ± 0.017	0.944 ± 0.024	0.886 ± 0.050	0.629 ± 0.056	0.857
		FlowRL	0.964 ± 0.014	0.910 ± 0.011	0.870 ± 0.018	0.452 ± 0.036	0.799
	Target Matching	FLAG (Ours)	0.974 ± 0.016	0.950 ± 0.011	0.912 ± 0.044	0.680 ± 0.062	0.879
			MyoSuite (4)				
Policy	**Method**	**Alg.**	**Reach-Hard**	**Obj-Hold-Hard**	**Key-Turn-Hard**	**Pen-Twirl-Hard**	**Avg.**
Gauss.	-	CrossQ	0.600 ± 0.217	1.000 ± 0.100	0.900 ± 0.150	0.100 ± 0.133	0.650
Express.	Action Gradient	DIPO	0.600 ± 0.219	0.340 ± 0.201	0.560 ± 0.408	0.760 ± 0.174	0.565
		QSM	0.089 ± 0.069	0.000 ± 0.000	0.209 ± 0.153	0.026 ± 0.048	0.081
	BPTT-based Actor-Critic	DACERv2	0.800 ± 0.107	0.925 ± 0.157	0.911 ± 0.100	0.800 ± 0.129	0.859
		DIME	0.900 ± 0.167	1.000 ± 0.017	1.000 ± 0.000	0.700 ± 0.233	0.900
		FlowRL	0.000 ± 0.008	0.017 ± 0.042	0.000 ± 0.000	0.050 ± 0.067	0.017
	Target Matching	FLAG (Ours)	0.930 ± 0.078	1.000 ± 0.000	0.930 ± 0.142	0.870 ± 0.174	0.933

5.2 FLAG Outperforms BPTT-based Actor-Critic and Action Gradient Methods

FLAG does not rely on critic gradients, making it robust even when critic gradient signals are unreliable. To investigate this, we compare methods in two settings: with and without CrossQ. For BPTT baselines, we use DIME [11], FlowRL [26], and DACERv2 [48], and for action-gradient baselines, we use DIPO [49] and QSM [38]. Following Section 5.1, we match critics using CrossQ and distributional critics across all methods[4]. As shown in Table 2, FLAG outperforms all baselines across high-dimensional benchmarks. To further examine FLAG's robustness to critic quality, we conduct experiments without CrossQ (Figure 4) [5]. While gradient-based methods are susceptible to noisy signals from an insufficiently trained critic, FLAG aggregates critic signals through importance-weighted averaging rather than direct differentiation, smoothing out noise in the learning target and thereby maintaining stable performance.

Table 3: Ablation study on buffer size and covariance scheduling. We report returns normalized by 1k in DMC Dog-run tasks. † denotes the hyperparameters used in Section 5.1 and Section 5.2. All experiments are conducted with 5 random seeds. The highest and second scores are highlighted in each column, respectively.

	Buffer Size					**Covariance Scheduling σ_{init} (σ_{final})**							
	0	**10.24k†**	**51.2k**	**102.4k**	**204.8k**	**-1(-1)**	**-1(-2)**	**-1(-3)**	**-2(-2)**	**-2(-3)†**	**-2(-4)**	**-2(-5)**	**-2(-6)**
Return (1k) ↑	0.601	0.680	0.670	0.618	0.589	0.614	0.675	0.732	0.588	0.680	0.597	0.547	0.269

5.3 FLAG Key Design Choices Align with Theoretical Results

We empirically show that the key design choices described in Section 4.4 help control the terms in the monotonic improvement bound of Theorem 4.5. This bound identifies two controllable quantities in practice: (i) the covariance-dependent drift term and (ii) the CFM projection error ϵ_k^{proj}. Accordingly, FLAG uses covariance scheduling and a guidance buffer (Section 4.4). In this section, we ablate these two components to examine how they affect the corresponding terms in practice.

Covariance Scheduling The condition $\lambda > \alpha C_\Sigma \sigma_k^2$ becomes easier to satisfy as the local covariance decreases, since σ_k^2 directly suppresses the reward-drift term (Eq. (26)). However, an overly

[4]Since DACERv2 trains critic to predict the mean and standard deviation of the Q-value, we exclude distributional critics.
[5]We only include DACERv2 and DIME in this comparison, as these methods achieve comparable performance in Table 2.

small σ_k can restrict the proposal region, thus weaken the guidance signal. To balance these effects, FLAG anneals the local covariance from σ_{init} to σ_{final}. As shown in Table 3, moderate annealing yields improved performance by allowing early local exploration while gradually reducing the drift. In contrast, overly aggressive annealing shrinks the local search region too early, leading to convergence to suboptimal local modes and eventual policy collapse.

Guidance Buffer The CFM projection error ϵ_k^{proj} measures how accurately the flow policy realizes the improved target actions obtained from the M-step. The guidance buffer reduces this error by caching recently computed target actions and reusing them across multiple off-policy updates, thereby providing denser supervision for the CFM projection. As shown in Table 3, removing the guidance buffer degrades performance, suggesting insufficient projection of local improvements into the flow policy. Conversely, an excessively large buffer also hurts performance, as stale targets become increasingly mismatched with the current policy. A moderate buffer size achieves the best trade-off between supervision density and target freshness, keeping ϵ_k^{proj} small in practice.

6 Conclusion

We presented FLAG, a MaxEnt-RL framework that optimizes an expressive policy through a supervised learning paradigm via latent-augmented guidance. By leveraging the deterministic property of the flow map, FLAG replaces the global proposal distribution of prior importance sampling-based methods with a local one, avoiding both the weight degeneracy of global sampling and the numerical instability of BPTT. Empirically, FLAG scales to high-dimensional action spaces with limited Q-function evaluations, and performs competitively with methods that rely on BPTT and critic gradients. To the best of our knowledge, this is the first supervision-based approach to scale to such high-dimensional control environments. As a limitation, the combination of the base flow policy and local Gaussian head is one particular instantiation of our framework; a more principled construction may be obtained via ODE-to-SDE conversion [15], which we leave for future work.

References

[1] Abbas Abdolmaleki, Jost Tobias Springenberg, Jonas Degrave, Steven Bohez, Yuval Tassa, Dan Belov, Nicolas Heess, and Martin Riedmiller. Relative entropy regularized policy iteration. *arXiv preprint arXiv:1812.02256*, 2018.

[2] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

[3] Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):1–34, 2011.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. In *NeurIPS Deep Learning Symposium*, 2016.

[5] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458, 2017.

[6] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[7] Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. CrossQ: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. In *International Conference on Learning Representations*, 2024.

[8] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[10] Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. MyoSuite: A contact-rich simulation suite for musculoskeletal motor control. In *Learning for Dynamics and Control*, pages 492–507, 2022.

[11] 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 *International Conference on Machine Learning*, pages 6958–6977, 2025.

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

[13] 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, pages 53945–53968, 2024. doi: 10.52202/079017-1708.

[14] 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. In *Advances in Neural Information Processing Systems*, 2025.

[15] Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, 2025.

[16] Xiaoyi Dong, Jian Cheng, and Xi Sheryl Zhang. Maximum entropy reinforcement learning with diffusion policy. In *International Conference on Machine Learning*, pages 13963–13983, 2025.

[17] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 20132–20145, 2021.

[18] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.

[19] 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, 2017.

[20] 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*, pages 1861–1870, 2018.

[21] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, 1989.

[22] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[23] Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. In *International Conference on Machine Learning*, pages 33352–33403, 2025.

[24] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[25] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

[26] Lei Lv, Yunfei Li, Yu Luo, Fuchun Sun, Tao Kong, Jiafeng Xu, and Xiao Ma. Flow-based policy for online reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

[27] Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. In *International Conference on Machine Learning*, pages 41837–41853, 2025.

[28] David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations*, 2026.

[29] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[30] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[31] Michal Nauman and Marek Cygan. On the theory of risk-aware agents: Bridging actor-critic and economics. In *ICML Workshop on Aligning Reinforcement Learning Experimentalists and Theorists*, 2024.

[32] Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: Scaling for compute and sample efficient continuous control. In *Advances in Neural Information Processing Systems*, volume 37, pages 113038–113071, 2024.

[33] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.

[34] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[36] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[37] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *International Conference on Machine Learning*, pages 745–750, 2007.

[38] Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via Q-score matching. In *International Conference on Machine Learning*, pages 41163–41182, 2024.

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

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

[41] Alexander Shapiro. Monte carlo sampling methods. *Handbooks in Operations Research and Management Science*, 10:353–425, 2003.

[42] Philip Thomas and Emma Brunskill. Importance sampling with unequal support. In *AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[43] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

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

[45] Marc Toussaint. Robot trajectory optimization using approximate inference. In *International Conference on Machine Learning*, pages 1049–1056, 2009.

[46] Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

[47] 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. In *Advances in Neural Information Processing Systems*, volume 37, pages 54183–54204, 2024. doi: 10.52202/079017-1717.

[48] 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.

[49] 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.

[50] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, volume 22, pages 1433–1438, 2008.

A Computational Challenges in Flow-based MaxEnt-RL

In this section, we describe in detail the technical challenges of applying flow-based policies within the standard MaxEnt-RL framework. Specifically, we define our policy $\tilde{\pi}_\theta$ as a flow-induced distribution where an action is generated by solving the ODE $T_\theta(s, z) = a^1 = z + \int_0^1 v_\theta(a^\tau, \tau, s)d\tau$ following Eq. (6).

Unbiased log-density estimation using Hutchinson trace estimator. A significant advantage of flow-based policies is the ability to recover exact log densities, which are typically unavailable in diffusion models. For a flow policy defined by the ODE trajectory ϕ_t^τ, the log-density admits a path-integral form:

$$\log \tilde{\pi}_\theta(a_t \mid s_t) = \log p_z(z_t) - \int_0^1 \mathrm{tr}\Big(J_\theta^\tau\Big)\, d\tau, \qquad J_\theta^\tau \triangleq \frac{\partial u_\theta^\tau}{\partial \phi}\big(s_t, \phi_t^\tau\big).$$

Hutchinson's identity [21] allows us to estimate the trace of a square matrix J. Using any random vector ϵ s.t. $\mathbb{E}[\epsilon\epsilon^\top] = I$, we obtain the unbiased estimator, $\mathrm{Tr}(J) = \mathbb{E}_\epsilon[\epsilon^\top J\epsilon]$. Applying this estimator to $\log \tilde{\pi}_\theta(a \mid s)$ yields the following.

$$\log \tilde{\pi}_\theta(a_t \mid s_t) = \mathbb{E}_\epsilon\left[\log p_z(z_t) - \int_0^1 \epsilon^\top J_\theta^\tau \epsilon\, d\tau\right]. \tag{28}$$

In practice, the quadratic form $\epsilon^\top J_\theta^\tau \epsilon$ is computed without forming J_θ^τ explicitly leveraging Jacobian-vector product. With reparameterization $a_t = T_\theta(s_t, z_t)$ and Eq. (28), we can form an unbiased Monte Carlo estimator of MaxEnt objective as follows:

$$\mathcal{J}_{\text{MaxEnt}}(\theta) = \mathbb{E}_{z_t \sim p_z,\, \epsilon}\left[Q_\phi\big(s_t, T_\theta(s_t, z_t)\big) - \log p_z(z_t) - \int_0^1 \epsilon^\top J_\theta^\tau \epsilon\Big)\right]. \tag{29}$$

Pathwise gradient requires differentiating through the entire ODE. Although Eq. (29) enables unbiased *evaluation*, optimizing θ by a direct pathwise gradient requires $\partial a_t/\partial\theta$. Let $S_t^\tau \triangleq \frac{\partial \phi_t^\tau}{\partial \theta}$ denote the parameter sensitivity along the flow. Differentiating the ODE gives the sensitivity equation

$$\frac{dS_t^\tau}{d\tau} = J_\theta^\tau\, S_t^\tau + \frac{\partial u_\theta^\tau}{\partial \theta}\big(s_t, \phi_t^\tau\big), \qquad S_t^0 = 0, \qquad \frac{\partial a_t}{\partial \theta} = S_t^1. \tag{30}$$

Therefore, the pathwise gradient of the Q-term is

$$\nabla_\theta Q_\phi\big(s_t, a_t\big) = \nabla_a Q_\phi(s_t, a)\big|_{a=a_t}\, S_t^1. \tag{31}$$

Computing S_t^1 entails differentiating through the entire ODE evolution in $\tau \in [0, 1]$. Since the sensitivity dynamics repeatedly apply the local Jacobian J_θ^τ along the trajectory, the accumulated transformation can become ill-conditioned, leading to vanishing/exploding sensitivities [6, 34].

The log-density gradient involves second-order terms (HVP). The Hutchinson integrand depends on the state through $J_\theta^\tau = \partial u_\theta^\tau/\partial\phi$. Its state gradient is

$$\nabla_{\phi_t^\tau}\big(\epsilon^\top J_\theta^\tau \epsilon\big) = \Big\langle \epsilon, \Big(\nabla_\phi^2 u_\theta^\tau(s_t, \phi_t^\tau)\Big)\epsilon \Big\rangle,$$

which is a Hessian–vector product (a double contraction with ϵ). Propagating this along the flow yields the score term

$$\nabla_a \log \tilde{\pi}_\theta(a_t \mid s_t) = J_0^\top \nabla_z \log p_z(z_t) - \mathbb{E}_\epsilon\Bigg[\int_0^1 (J_\theta^\tau)^\top \underbrace{\Big\langle \epsilon, \Big(\nabla_\phi^2 u_\theta^\tau(s_t, \phi_t^\tau)\Big)\epsilon \Big\rangle}_{\text{HVP}}\, d\tau\Bigg], \tag{32}$$

where $\nabla_a = (J_\theta^\tau)^\top \nabla_{\phi_t^\tau}$.

Implication. Eq. (30) and Eq. (31) show that even the Q-term requires differentiating through the entire ODE. More importantly, Eq. (32) reveals that the entropy gradient additionally involves Hessian–vector products of the vector field, making direct MaxEnt-RL optimization with flow policies more expensive than objective evaluation.

B Derivations

In this section, we make detail derivations in Section 4.

B.1 Global and Local Consistency

B.1.1 Marginal Distributions Consistency (Corollary 4.2)

While our general framework assumes an infinite horizon, for notational clarity, we restrict our attention to a finite horizon T in this derivation. Fix an initial pair (s_t, a_t), and any z_t; write $\hat{s}_t = (s_t, z_t)$. Consider the suffix trajectories

$$\tau = (s_{t+1}, a_{t+1}, \ldots, s_T, a_T), \qquad \hat{\tau} = (\hat{s}_{t+1}, a_{t+1}, \ldots, \hat{s}_T, a_T).$$

Under the augmented process with z-conditioned policy $\hat{\pi}(\cdot \mid \hat{s})$ and i.i.d. z's,

$$p_{\hat{\pi}}\left(\hat{\tau} \mid \hat{s}_t, a_t\right) = \prod_{i=t}^{T-1} p(s_{i+1} \mid s_i, a_i)\, p_z(z_{i+1})\, \hat{\pi}(a_{i+1} \mid s_{i+1}, z_{i+1}). \tag{33}$$

Marginalizing out $z_{t+1:T}$ gives the (s, a)–trajectory law

$$\begin{aligned}\int p_{\hat{\pi}}(\hat{\tau} \mid \hat{s}_t, a_t) \prod_{i=t+1}^{T-1} dz_i &= \prod_{i=t}^{T-1} p(s_{i+1} \mid s_i, a_i) \underbrace{\int p_z(z_{i+1})\, \hat{\pi}(a_{i+1} \mid s_{i+1}, z_{i+1})\, dz_{i+1}}_{=\,\pi(a_{i+1} \mid s_{i+1})} \\ &= \prod_{i=t}^{T-1} p(s_{i+1} \mid s_i, a_i)\, \pi(a_{i+1} \mid s_{i+1}) =: p_\pi(\tau \mid s_t, a_t).\end{aligned} \tag{34}$$

Thus, after integrating out z, the (s, a)–trajectory distribution exactly matches that induced by the marginal policy π. We can extend this result to the infinite horizon as $T \to \infty$.

Lemma B.1. *In the z-MDP framework, let $\hat{\pi}$ be the local policy defined in Eq. (8) and π be the global policy defined in Eq. (9). The discounted state and state–action marginal distributions induced by $\hat{\pi}$ satisfy:*

$$\hat{\rho}_{\hat{\pi}}(s, z, a) \;=\; \rho_\pi(s)\, p_z(z)\, \hat{\pi}(a \mid s, z), \qquad \hat{\rho}_{\hat{\pi}}(\hat{s}) \;=\; \rho_\pi(s)\, p_z(z). \tag{35}$$

Proof. Consider the joint probability of the state and latent variable at time step t. Since z_t is sampled i.i.d. from $p_z(\cdot)$ and is independent of s_t, we have $P_{\hat{\pi}}(z_t = z \mid s_t = s) = p_z(z)$. Furthermore, leveraging the trajectory consistency in Eq. (34), the marginal distribution of s_t under $\hat{\pi}$ is identical to that under π, i.e., $P_{\hat{\pi}}(s_t = s) = P_\pi(s_t = s)$.

Combining these observations, the joint probability at time t factorizes as:

$$P_{\hat{\pi}}(s_t = s, z_t = z) \;=\; P_\pi(s_t = s)\, p_z(z). \tag{36}$$

Substituting this equality into the definition of the discounted state marginal distribution yields:

$$\begin{aligned}\hat{\rho}_{\hat{\pi}}(s, z) &= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t P_{\hat{\pi}}(s_t = s, z_t = z) \\ &= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t P_\pi(s_t = s)\, p_z(z) \\ &= \left[(1-\gamma) \sum_{t=0}^{\infty} \gamma^t P_\pi(s_t = s)\right] p_z(z) \\ &= \rho_\pi(s)\, p_z(z).\end{aligned} \tag{37}$$

Finally, for the state–action marginal distribution, the following equation concludes the proof:

$$\hat{\rho}_{\hat{\pi}}(s, z, a) \;=\; \big[\rho_\pi(s)\, p_z(z)\big] \hat{\pi}(a \mid s, z), \qquad \rho_\pi(s, a) = \int p_z(z)\, \hat{\pi}(a \mid s, z)\, \rho_\pi(s)\, dz. \tag{38}$$

□

B.1.2 Q-function Consistency (Corollary 4.4)

Now we compare soft returns. The augmented definition is

$$Q^{\hat{\pi}}(\hat{s}_t, a_t) = \mathbb{E}_{\hat{\tau}\sim p_{\hat{\pi}}}\left[\sum_{k=0}^{\infty}\gamma^k\left(\hat{r}(\hat{s}_{t+k}, a_{t+k}) + \alpha\,\mathcal{H}\left(\hat{\pi}(\cdot \mid \hat{s}_{t+k}), \tilde{\pi}(\cdot \mid s_{t+k})\right)\right)\right].$$

Taking expectation over $z_{t+1:\infty}$ and using Eq. (34), the reward terms coincide by $\hat{r}(\hat{s}, a) = r(s, a)$, and the cross-entropy terms reduce to

$$\mathbb{E}_{z_{t+k}\sim p_z}\left[H\left(\hat{\pi}(\cdot \mid \hat{s}_{t+k}), \tilde{\pi}(\cdot \mid s_{t+k})\right)\right] = \mathcal{H}\left(\pi(\cdot \mid s_{t+k}), \tilde{\pi}(a \mid s_{t+k})\right).$$

Hence, marginalizing over $z_{t+1:\infty}$ gives

$$Q^{\pi}(\hat{s}_t, a_t) = \mathbb{E}_{\tau\sim p_\pi}\left[\sum_{k=0}^{\infty}\gamma^k\Big(r(s_{t+k}, a_{t+k}) + \alpha\,\mathcal{H}\big(\pi(\cdot \mid s_{t+k}), \tilde{\pi}(a \mid s_{t+k})\big)\Big)\right] =: Q^{\pi}(s_t, a_t).$$

The right–hand side does not depend on the initial z_t, therefore $Q^{\hat{\pi}}(s_t, z_t, a_t) = Q^{\pi}(s_t, a_t)$ is valid for all z_t.

B.2 EM algorithm Derivations (Section 4.3)

This section provides the full derivation of FLAG's EM update in Section 4.3. We proceed in three steps: (i) we cast cross-entropy augmented RL objective as probabilistic inference and derive the variational lower bound both in the original MDP and the z-MDP; (ii) we solve the E-step in closed form, yielding a non-parametric target distribution q_k; (iii) we project q_k back to the parametric policy class through the M-step. Throughout, we follow the MPO [2] framework while adapting it to our cross-entropy augmented reward structure, which we discuss in the remark below.

An infinite-horizon discounted reward formulation can be cast as inference problem as follows:

$$p\left(\mathcal{O} = 1 \mid \tau\right) \propto \exp\left(\sum_{t=1}^{\infty}\gamma^t\left(r_t/\lambda\right)\right) \tag{39}$$

where $r_t = r(s_t, a_t)$. Here, the optimality variable $\mathcal{O}$ reveals *the event of obtaining maximum reward by choosing an action*. We define the augmented reward in the spirit of Haarnoja et al. [20].

$$\begin{aligned} r_\pi(s_t, a_t) &\triangleq r(s_t, a_t) + \alpha\mathbb{E}_{s_{t+1}\sim p(\cdot|s_t,a_t)}\left[\mathcal{H}\left(\pi(\cdot \mid s_{t+1}), \tilde{\pi}(\cdot \mid s_{t+1})\right)\right], \\ \hat{r}_{\hat{\pi}}(\hat{s}_t, a_t) &\triangleq r(\hat{s}_t, a_t) + \alpha\mathbb{E}_{\hat{s}_{t+1}\sim \hat{p}(\cdot|\hat{s}_t,a_t)}\left[\mathcal{H}\left(\hat{\pi}(\cdot \mid \hat{s}_{t+1}), \tilde{\pi}(\cdot \mid s_{t+1})\right)\right]. \end{aligned} \tag{40}$$

Remark. **Remark on the reward structure and the E-step.** The augmented reward in Eq. (40) is defined separately for each policy (r_π for π, r_q for q), since the cross-entropy bonus follows the soft Bellman structure of SAC [20]. Although this augmentation is policy-dependent, it is compatible with the MPO-style E-step derivation. Following MPO [2], the E-step starts from the reference $q = \hat{\pi}_k$ and performs a one-step lookahead, so the Q-function used inside the E-step is $Q^{\hat{\pi}_k}$, in which the entropy bonus for $\hat{\pi}_k$ is already absorbed. The resulting energy

$$f_{\hat{s},k}(a) = Q^{\hat{\pi}_k}(\hat{s}, a) - \alpha\log\tilde{\pi}_{\theta_k}(a \mid s) = Q^{\pi_k}(s, a) - \alpha\log\tilde{\pi}_{\theta_k}(a \mid s)$$

is a function of (s, a) alone, with θ_k treated as a constant within iteration k. Hence the MPO closed-form E-step solution and the subsequent M-step derivation apply without modification. The effect of the parameter update $\theta_k \to \theta_{k+1}$ between iterations (the reward drift) is analyzed separately in the monotonic-improvement proof (Appendix C.3).

To derive a tractable objective, we construct a variational lower bound on the log-likelihood of optimality $\log p_\pi(\mathcal{O} = 1)$, following the RL-as-inference framework [24]. With the policy induced trajectory $\tau \sim p_\pi(\tau)$ in Eq. (34), we apply Jensen's inequality with an auxiliary distribution $p_q(\tau)$:

$$\begin{aligned} \log p_\pi(\mathcal{O} = 1) &= \log\int p_\pi(\tau)p(\mathcal{O} = 1 \mid \tau)d\tau \geq \int p_q(\tau)\left[\log p(\mathcal{O} = 1 \mid \tau) + \log\frac{p_\pi(\tau)}{p_q(\tau)}\right]d\tau \\ &= \mathbb{E}_q\left[\sum_t \gamma^t r_q(s_t, a_t)/\lambda\right] - D_{\mathrm{KL}}\left(p_q(\tau) \,\|\, p_\pi(\tau)\right), \end{aligned}$$

Following the standard derivation in [2, 24], we absorb the entropy component of the trajectory-level KL into the per-step augmented reward

$$r_q(s_t, a_t) = r(s_t, a_t) + \alpha\, \mathbb{E}_{s_{t+1}}\big[\mathcal{H}(q(\cdot \mid s_{t+1}), \tilde{\pi}(\cdot \mid s_{t+1}))\big],$$

and express the remaining per-step KL in discounted form, yielding the variational objective

$$J(q, \xi) = \mathbb{E}_q\left[\sum_{t=0}^{\infty} \gamma^t \left(\frac{r_q(s_t, a_t)}{\lambda} - D_{\mathrm{KL}}(q(\cdot \mid s_t) \,\|\, \pi(\cdot \mid s_t; \xi))\right)\right].$$

We now extend the same variational argument to the z-MDP $\hat{\mathcal{M}}$, on which FLAG actually operates. Replacing the trajectory τ with the augmented trajectory $\hat{\tau}$ and using the marginal consistency established in Corollary 4.2, we obtain:

$$\begin{aligned}
\log p_{\hat{\pi}}(\mathcal{O} = 1) &= \log \int p_{\hat{\pi}}(\hat{\tau}) p(\mathcal{O} = 1 \mid \hat{\tau}) d\hat{\tau} \geq \int p_q(\hat{\tau}) \left[\log p(\mathcal{O} = 1 \mid \hat{\tau}) + \log \frac{p_{\hat{\pi}}(\hat{\tau})}{p_q(\hat{\tau})}\right] d\hat{\tau} \\
&= \mathbb{E}_q\left[\sum_t \hat{r}_q(\hat{s}_t, a_t)/\lambda\right] - D_{\mathrm{KL}}\left(p_q(\hat{\tau}) \parallel p_{\hat{\pi}}(\hat{\tau})\right) \\
\therefore\ J(q, \xi) &= \mathbb{E}_q\left[\sum_{t=0}^{\infty} \gamma^t \left[\hat{r}_q(\hat{s}_t, a_t) - \lambda D_{\mathrm{KL}}\left(q(\cdot \mid \hat{s}_t) \parallel \hat{\pi}(\cdot \mid \hat{s}_t, \xi)\right)\right]\right].
\end{aligned}$$

Constrained E-step We follow the MPO [2] derivation and adapt it to hard KL constraints and one-step bootstrapping. At iteration k, the objective function is

$$\max_q \mathbb{E}_{\hat{s} \sim \hat{\rho}_{\hat{\pi}}}\left[\mathbb{E}_{a \sim q(\cdot \mid \hat{s})}\left[Q^{\pi}(s, a) - \alpha \log \tilde{\pi}(a \mid s)\right]\right] \qquad s.t.\ \left[D_{\mathrm{KL}}\left(q(\cdot \mid \hat{s}) \parallel \hat{\pi}(\cdot \mid \hat{s}; \xi_k)\right) \leq \epsilon. \right. \tag{41}$$

This objective has closed-form solution q_k, which is called the non-parametric variational distribution:

$$q_k(a \mid \hat{s}) = \frac{\hat{\pi}(a \mid \hat{s}; \xi_k) \exp\left(f_{\hat{s},k}(a)/\lambda^*\right)}{Z(\hat{s})}, \qquad Z(\hat{s}) = \int \hat{\pi}(a \mid \hat{s}; \xi_k) \exp\left(f_{\hat{s},k}(a)/\lambda^*\right)\, da. \tag{42}$$

M-step In the M-step, given the fixed non-parametric target distribution q_k, we update the policy parameters ξ_k by projecting q_k back to our parametric rollout policy class:

$$\xi_{k+1} \in \arg\min_{\xi} \mathbb{E}_{\hat{\rho}_{\hat{\pi}_k}}\left[D_{\mathrm{KL}}(q_k(\cdot \mid \hat{s}) \,\|\, \hat{\pi}(\cdot \mid \hat{s}; \xi_k))\right]. \tag{43}$$

C Proofs

Standing Assumptions. We consider an infinite-horizon γ-discounted MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \rho_0, \gamma)$. Fix a reference measure μ on $\mathcal{A}$ and assume all policies admit densities with respect to μ. Throughout this appendix, we write da in place of $d\mu(a)$ for notational simplicity. Accordingly, $\tilde{\pi}(\cdot \mid s)$ and $\pi(\cdot \mid s)$ denote μ-densities, and all KL divergences and expectations are taken with respect to these densities. Furthermore, we assume that $\tilde{\pi}$ and π are differentiable in $\mathcal{A}$.

Assumption C.1 (Bounded Augmented Reward)**.** The entropy-augmented reward is uniformly bounded. Specifically, there exists a constant $R_{\max}^{\mathrm{aug}} < \infty$ such that

$$\left|r(s, a) + \alpha\, b(s, a)\right| \leq R_{\max}^{\mathrm{aug}}, \qquad \forall (s, a), \tag{44}$$

where $b(s, a)$ denotes the entropy or cross-entropy bonus term (e.g., $-\log \tilde{\pi}(a \mid s)$).

Assumption C.2 (Regularity of Base Density)**.** For each state s, the base density $\tilde{\pi}(\cdot \mid s)$ is three-times continuously differentiable with respect to a and is strictly positive μ-a.e. Moreover, the magnitudes of its third-order partial derivatives are pointwise bounded by a nonnegative

function $M_s(a)$ satisfying the integrability condition:

$$\int_{\mathcal{A}} M_s(a)\, da < \infty. \tag{45}$$

Assumption C.3 (Integrable gradient of the base density)**.** For each state s, the gradient of the base density $\tilde{\pi}(\cdot \mid s)$ $w.r.t$ a is integrable:

$$\|\nabla_a \tilde{\pi}(\cdot \mid s)\|_{L^1} \triangleq \int_{\mathcal{A}} \|\nabla_a \tilde{\pi}(a \mid s)\|_2\, da < \infty. \tag{46}$$

This condition ensures that the density $\tilde{\pi}$ does not exhibit unbounded variation in the action space.

Assumption C.4 (Finite Partition Function)**.** For every state $\hat{s}$ and iteration k, the function $Q^k(\hat{s}, \cdot)$ is measurable, and the corresponding Boltzmann distribution is well-defined (i.e., normalizable):

$$\int_{\mathcal{A}} \exp\left(\frac{Q^k(\hat{s}, a)}{\alpha}\right) da < \infty. \tag{47}$$

C.1 Proof of Proposition 4.3

In this sectionassume that the Gaussian smoothing covariance is state-independent and isotropic. For notational simplicity, we denote it by Σ, with $\Sigma = \sigma^2 I$ and denote $\tilde{\pi}_\theta$ simply as $\tilde{\pi}$ hereafter. We prove Proposition 4.3 in two parts: Lemma C.5 establishes the TV bound, and Lemma C.6 establishes the second-order KL bound.

Let $\delta \sim \mathcal{N}(0, \Sigma)$ be an independent noise at $a = T_\theta(s, z) + \delta$. Mathematically, this takes the form

$$\pi(\cdot \mid s) = \tilde{\pi}(\cdot \mid s) * \varphi_\Sigma, \tag{48}$$

where $*$ is the convolution sign in the action space.

C.1.1 Total variation distance bound

We assume that the following inequality holds for all s:

$$\|\nabla_a \tilde{\pi}(\cdot \mid s)\|_{L^1} \triangleq \int \|\nabla_a \tilde{\pi}(a \mid s)\|_2\, da < \infty. \tag{49}$$

Lemma C.5. *Suppose the global policy admits a Gaussian-kernel convolution form*

$$\pi(a \mid s) = \int \tilde{\pi}(a - \delta \mid s)\, \varphi_\Sigma(\delta)\, d\delta, \tag{50}$$

where φ_Σ is the density of $\mathcal{N}(0, \Sigma)$. Then the following inequality holds for all $s \in \mathcal{S}$

$$D_{\mathrm{TV}}\left(\pi(\cdot \mid s), \tilde{\pi}(\cdot \mid s)\right) \le \frac{1}{2} \|\nabla_a \tilde{\pi}(\cdot \mid s)\|_{L^1}\, \mathbb{E}_{\delta \sim \mathcal{N}(0,\Sigma)}\left[\|\delta\|_2\right]. \tag{51}$$

Proof. By definition of the convolution in Eq. (50),

$$\pi(a \mid s) - \tilde{\pi}(a \mid s) = \int \left(\tilde{\pi}(a - \delta \mid s) - \tilde{\pi}(a \mid s)\right) \varphi_\Sigma(\delta)\, d\delta. \tag{52}$$

Using the triangle inequality,

$$\|\pi(\cdot \mid s) - \tilde{\pi}(\cdot \mid s)\|_{L^1} = \int \left| \int \left(\tilde{\pi}(a - \delta \mid s) - \tilde{\pi}(a \mid s)\right) \varphi_\Sigma(\delta)\, d\delta \right| da \tag{53}$$

$$\le \int \int \left|\tilde{\pi}(a - \delta \mid s) - \tilde{\pi}(a \mid s)\right| \varphi_\Sigma(\delta)\, d\delta\, da \tag{54}$$

Since the integrand is nonnegative, Tonelli's theorem yields

$$\|\pi(\cdot \mid s) - \tilde{\pi}(\cdot \mid s)\|_{L^1} \le \int \varphi_\Sigma(\delta) \left(\int \left|\tilde{\pi}(a - \delta \mid s) - \tilde{\pi}(a \mid s)\right| da \right) d\delta. \tag{55}$$

For each fixed (a, δ), by the integral form of the mean value theorem,

$$\tilde{\pi}(a - \delta \mid s) - \tilde{\pi}(a \mid s) = -\int_0^1 \delta^\top \nabla_a \tilde{\pi}(a - t\delta \mid s)\, dt. \tag{56}$$

Applying the Cauchy–Schwarz inequality yields

$$|\tilde{\pi}(a - \delta \mid s) - \tilde{\pi}(a \mid s)| \le \|\delta\|_2 \int_0^1 \|\nabla_a \tilde{\pi}(a - t\delta \mid s)\|_2\, dt. \tag{57}$$

Integrating over a and using the change of variables $u = a - t\delta$,

$$\begin{aligned}\int |\tilde{\pi}(a - \delta \mid s) - \tilde{\pi}(a \mid s)|\, da &\le \|\delta\|_2 \int_0^1 \int \|\nabla_a \tilde{\pi}(a - t\delta \mid s)\|_2\, da\, dt \\ &= \|\delta\|_2 \int_0^1 \int \|\nabla_a \tilde{\pi}(u \mid s)\|_2\, du\, dt \\ &= \|\delta\|_2 \|\nabla_a \tilde{\pi}(\cdot \mid s)\|_{L^1}. \end{aligned} \tag{58}$$

Plugging this into Eq. (55) bound gives

$$\boxed{\|\pi(\cdot \mid s) - \tilde{\pi}(\cdot \mid s)\|_{L^1} \le \|\nabla_a \tilde{\pi}(\cdot \mid s)\|_{L^1}\ \mathbb{E}_{\delta \sim \mathcal{N}(0,\Sigma)}[\|\delta\|_2].}$$

Finally, using $D_{\mathrm{TV}}(p, q) = \frac{1}{2}\|p - q\|_{L^1}$ concludes the proof. Moreover, by Cauchy–Schwarz,

$$\mathbb{E}_{\delta \sim \mathcal{N}(0,\Sigma)}[\|\delta\|_2] \le \sqrt{\mathbb{E}[\|\delta\|_2^2]} = \sqrt{\mathbb{E}[\delta^\top \delta]} = \sqrt{\mathrm{tr}(\Sigma)}.$$

Combining the above bounds implies $D_{\mathrm{TV}}(\pi, \tilde{\pi}) = \mathcal{O}\big(\sqrt{\mathrm{tr}(\Sigma)}\big)$ as $\mathrm{tr}(\Sigma) \to 0$.. □

C.1.2 Kullback-Leibler Divergence Bound

In this section, we show that when $\pi(\cdot \mid s)$ is obtained by smoothing $\tilde{\pi}(\cdot \mid s)$ with a *small-variance* Gaussian kernel in Eq. (48), the discrepancy term $D_{\mathrm{KL}}(\pi \parallel \tilde{\pi})$ vanishes at second order in $\mathrm{tr}(\Sigma)$.

Recall the identity

$$\mathcal{H}(\pi, \tilde{\pi}) = \mathcal{H}(\pi) + D_{\mathrm{KL}}(\pi \parallel \tilde{\pi}),$$

where $\mathcal{H}(\pi, \tilde{\pi}) \triangleq \mathbb{E}_\pi[-\log \tilde{\pi}]$ is the cross-entropy.

Lemma C.6 (Second-order KL bound under small Gaussian smoothing)**.** *For each state s, let $\tilde{\pi}(\cdot \mid s)$ satisfy Assumption C.2 and strictly positive in a. Assume that the global policy admits the Gaussian smoothing form in Eq. (50). In addition, we assume the integrability condition*

$$\int \frac{\big(\mathrm{tr}(\Sigma \nabla_a^2 \tilde{\pi}(a \mid s))\big)^2}{\tilde{\pi}(a \mid s)}\, da < \infty. \tag{59}$$

Then, as $\mathrm{tr}(\Sigma) \to 0$,

$$D_{\mathrm{KL}}(\pi(\cdot \mid s) \,\|\, \tilde{\pi}(\cdot \mid s)) = \mathcal{O}\big(\mathrm{tr}(\Sigma)^2\big). \tag{60}$$

Consequently,

$$\mathcal{H}(\pi, \tilde{\pi}) = \mathcal{H}(\pi) + \mathcal{O}\big(\mathrm{tr}\,(\Sigma)^2\big),$$

i.e., the cross-entropy approximates the true entropy up to a second-order error in $\mathrm{tr}(\Sigma)$.

Proof. The smoothing form Eq. (50) can be written as an expectation over the Gaussian perturbation:

$$\pi(a \mid s) = \mathbb{E}_{\delta \sim \mathcal{N}(0,\Sigma)}\left[\tilde{\pi}(a - \delta \mid s)\right]. \tag{61}$$

When Σ is small, the random shift δ concentrates near 0. We therefore Taylor-expand $\tilde{\pi}(\cdot \mid s)$ around a and evaluate at $a - \delta$:

$$\tilde{\pi}(a - \delta \mid s) = \tilde{\pi}(a \mid s) - \nabla_a \tilde{\pi}(a \mid s)^\top \delta + \frac{1}{2}\delta^\top \nabla_a^2 \tilde{\pi}(a \mid s)\, \delta + \mathcal{O}(\|\delta\|_2^3). \tag{62}$$

Taking expectation of Eq. (62) w.r.t. $\delta \sim \mathcal{N}(0, \Sigma)$ yields:

$$\pi(a \mid s) = \tilde{\pi}(a \mid s) - \mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma)}\left[\nabla_a \tilde{\pi}(a \mid s)^\top \delta\right] + \mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma)}\left[\frac{1}{2}\delta^\top \nabla_a^2 \tilde{\pi}(a \mid s)\, \delta\right] + \mathcal{O}(\|\delta\|_2^3).$$

The first-order term vanishes because $\mathbb{E}[\delta] = 0$, and the second-order term becomes a trace term because $\mathbb{E}[\delta\delta^\top] = \Sigma$:

$$\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma)}\left[\delta^\top \nabla_a^2 \tilde{\pi}(a \mid s)\, \delta\right] = \mathrm{tr}\big(\Sigma \nabla_a^2 \tilde{\pi}(a \mid s)\big)\,.$$

Therefore, a third-order Taylor expansion yields

$$\pi(a \mid s) = \tilde{\pi}(a \mid s) + \frac{1}{2}\,\mathrm{tr}\Big(\Sigma \nabla_a^2 \tilde{\pi}(a \mid s)\Big) + R_3(a, s), \tag{63}$$

where the remainder satisfies $|R_3(a, s)| \le C(s)\,\mathbb{E}[\|\delta\|_2^3]$. Applying the Cauchy–Schwarz inequality and Isserlis' theorem for Gaussian moments, we obtain

$$\mathbb{E}[\|\delta\|_2^3] \le \big(\mathbb{E}[\|\delta\|_2^2]\big)^{1/2}\big(\mathbb{E}[\|\delta\|_2^4]\big)^{1/2} = \mathcal{O}\Big(\mathrm{tr}(\Sigma)^{1/2} \cdot \mathrm{tr}(\Sigma)\Big) = \mathcal{O}\Big(\mathrm{tr}(\Sigma)^{3/2}\Big)\,,$$

implying that the remainder term is of order $\mathcal{O}(\mathrm{tr}(\Sigma)^{3/2})$. Note that the remainder's contribution to $(\pi - \tilde{\pi})$ at order $\mathcal{O}(\mathrm{tr}(\Sigma)^{3/2})$ becomes $\mathcal{O}(\mathrm{tr}(\Sigma)^3)$ after squaring.

To establish the order of the KL divergence, we analyze the relative perturbation of the rollout policy π with respect to the base density $\tilde{\pi}$:

$$x(a) \triangleq \frac{\pi(a \mid s) - \tilde{\pi}(a \mid s)}{\tilde{\pi}(a \mid s)}. \tag{64}$$

By construction, $\pi(\cdot \mid s)$ is obtained by convolving $\tilde{\pi}(\cdot \mid s)$ with a small-variance Gaussian kernel. In addition, we assume a uniform relative closeness condition: for sufficiently small Σ,

$$\|x\|_\infty \le c\ \mathrm{tr}(\Sigma) \tag{65}$$

for some constant $c > 0$, which is chosen arbitrarily. Hence, $|x(a)| \le 1/2$ for all $a \in \mathcal{A}$ whenever $\mathrm{tr}(\Sigma) \le (2c)^{-1}$, justifying the uniform Taylor expansion

$$\log(1 + x) = x - \frac{x^2}{2} + R_3(x), \qquad |R_3(x)| \le C|x|^3 \ \text{ for } \ |x| \le \tfrac{1}{2}. \tag{66}$$

Using $\pi = \tilde{\pi}(1 + x)$, we expand

$$\begin{aligned} D_{\mathrm{KL}}(\pi \,\|\, \tilde{\pi}) &= \int \pi(a \mid s) \log \frac{\pi(a \mid s)}{\tilde{\pi}(a \mid s)}\, da \\ &= \int \tilde{\pi}(a \mid s)(1 + x(a)) \log(1 + x(a))\, da \\ &= \int \tilde{\pi}(a \mid s) \left(x(a) + \frac{x(a)^2}{2}\right) da + \int \tilde{\pi}(a \mid s)(1 + x(a)) R_3(x(a))\, da. \end{aligned} \tag{67}$$

The linear term vanishes since $\int \tilde{\pi} x\, da = \int (\pi - \tilde{\pi})\, da = 0$. For the remainder term, $|x| \le 1/2$ implies $1 + x \le 3/2$ and therefore

$$\left|\int \tilde{\pi}(1 + x) R_3(x)\, da\right| \le \frac{3C}{2} \int \tilde{\pi}|x|^3\, da \le \frac{3C}{2}\|x\|_\infty \int \tilde{\pi} x^2\, da.$$

Consequently,

$$D_{\mathrm{KL}}(\pi \,\|\, \tilde{\pi}) = \frac{1}{2}\int \tilde{\pi}(a \mid s) x(a)^2\, da + \mathcal{O}\left(\|x\|_\infty \int \tilde{\pi} x^2\, da\right). \tag{68}$$

Consequently, by Eq. (63) and Eq. (68), the leading term of $x(a)^2$ satisfies

$$x(a)^2 = \frac{1}{4}\left(\frac{\mathrm{tr}(\Sigma \nabla_a^2 \tilde{\pi}(a \mid s))}{\tilde{\pi}(a \mid s)}\right)^2 + \text{higher-order terms}.$$

Under the integrability condition in Eq. (59), this yields

$$\int \tilde{\pi} x^2\, da = \mathcal{O}\big(\mathrm{tr}(\Sigma)^2\big)\,.$$

Combining this with Eq. (65) and Eq. (68) proves $D_{\mathrm{KL}}(\pi \| \tilde{\pi}) = \mathcal{O}\left(\mathrm{tr}(\Sigma)^2\right)$. □

C.2 Monotonic Improvement by Soft Actor Critic

In this section, we establish the SAC perspective of Theorem 4.5, which connects FLAG's moment-matching update to SAC-style soft policy improvement. The argument proceeds in two steps. First, we show that under a sufficiently small local covariance Σ, the composite policy π remains close to the base flow policy $\tilde{\pi}$ in the sense of soft Q-function (Appendix C.2.1). This allows us to treat the SAC soft Bellman backup on $\tilde{\pi}$ as the reference dynamics for policy improvement. Second, we show that the moment-matching target $\mu_k^*(\hat{s})$ from Eq. (24) approximates the action gradient of the SAC objective $f_{\hat{s},k}$ via log-sum-exp trick, producing a zeroth-order BPTT-free estimate of the update direction that SAC realizes through the reparameterization-based action gradient (Appendix C.2.2).

C.2.1 Q-function Closeness between Composite Policy and Flow Policy

To show that the difference between the expected sum of returns with the policy π and those for the base flow policy $\tilde{\pi}$, first we need to look at how difference the discounted state distributions $\rho_\pi(s), \rho_{\tilde{\pi}}(s)$ are. This section refers to the proof of TRPO [39].

Lemma C.7. *The difference between the state distributions induced by π and $\tilde{\pi}$ has the following bound:*

$$\|\rho_\pi - \rho_{\tilde{\pi}}\|_1 \leq \frac{2\gamma\delta_{\mathrm{TV}}}{(1-\gamma)^2}, \tag{69}$$

where δ_{TV} is defined as

$$\delta_{\mathrm{TV}} \triangleq D_{\mathrm{TV}}^{\max}(\pi, \tilde{\pi}) = \sup_s D_{\mathrm{TV}}\left(\pi(\cdot \mid s), \tilde{\pi}(\cdot \mid s)\right) \leq \frac{1}{2}\left(\sup_s \|\nabla_a \tilde{\pi}(\cdot \mid s)\|_{L^1}\right) \cdot \sqrt{\mathrm{tr}\left(\Sigma\right)},$$

by leveraging Lemma C.5

Proof. Let d_t^π denote the state distribution in timestep t, then we can bound the total variation distance between d_t^π and $d_t^{\tilde{\pi}}$ as follows

$$D_{\mathrm{TV}}(d_t^\pi, d_t^{\tilde{\pi}}) \leq P(n_t > 0) \leq 1 - (1 - \delta_{\mathrm{TV}})^t \leq t\delta_{\mathrm{TV}}, \tag{70}$$

where n_t denote the number of times that $a_i \neq \tilde{a}_i$ for , i.e. the number of times that π and $\tilde{\pi}$ disagree before timestep t. See [39], Lemma 3 for more details.

The total variation distance between the discounted visitation measure between π and $\tilde{\pi}$ is represented with the state distribution d_t:

$$\|\rho_\pi - \rho_{\tilde{\pi}}\|_1 = \left\| \sum_{t=0}^{\infty} \gamma^t \left(d_t^\pi - d_t^{\tilde{\pi}}\right) \right\|_1. \tag{71}$$

By triangle inequality,

$$\|\rho_\pi - \rho_{\tilde{\pi}}\|_1 \leq \sum_{t=0}^{\infty} \gamma^t \left\|d_t^\pi - d_t^{\tilde{\pi}}\right\|_1. \tag{72}$$

Since $\|p - q\|_1 = 2D_{\mathrm{TV}}(p, q)$,

$$\|\rho_\pi - \rho_{\tilde{\pi}}\|_1 \leq \sum_{t=0}^{\infty} \gamma^t \left\|d_t^\pi - d_t^{\tilde{\pi}}\right\|_1 = 2\sum_{t=0}^{\infty} \gamma^t D_{\mathrm{TV}}\left(d_t^\pi, d_t^{\tilde{\pi}}\right) \tag{73}$$

$$= 2\sum_{t=0}^{\infty} \gamma^t \left(t\delta_{\mathrm{TV}}\right) = 2\gamma\delta_{\mathrm{TV}} \sum_{t=1}^{\infty} t\gamma^{t-1} = \frac{2\gamma\delta_{\mathrm{TV}}}{(1-\gamma)^2}. \tag{74}$$

□

This inequality states that if δ_{TV} is bounded $\delta_{\mathrm{TV}} = \mathcal{O}\left(\sqrt{\mathrm{tr}(\Sigma)}\right)$, then the difference of the discounted measure between π and $\tilde{\pi}$ is bounded with $\mathcal{O}\left(\sqrt{\mathrm{tr}(\Sigma)}\right)$.

Next, using Lemma C.7, we bound the Q-function discrepancy between the composite policy π and the base flow policy $\tilde{\pi}$ under their respective soft Bellman operators.

Lemma C.8 (Q-function closeness between π and $\tilde{\pi}$)**.** *Let Q^{π} and $Q^{\tilde{\pi}}$ denote the soft Q-functions associated with the composite policy π (under the **cross-entropy-regularized soft Bellman operator** $\mathcal{T}^{\pi}$) and the base flow policy $\tilde{\pi}$ (under the **entropy-regularized soft Bellman operator** $\mathcal{T}^{\tilde{\pi}}$), respectively. Under the standing assumptions,*

$$\|Q^{\pi} - Q^{\tilde{\pi}}\|_{\infty} \leq \frac{2\delta_{\mathrm{TV}}}{1-\gamma}\left(\alpha C_R + \frac{\gamma R_{\max}}{1-\gamma}\right) = \mathcal{O}\left(\frac{\sqrt{\mathrm{tr}(\Sigma)}}{(1-\gamma)^2}\right), \tag{75}$$

where $C_R = \sup_{s,a} |\log \tilde{\pi}(a \mid s)|$ and δ_{TV} is the maximum TV distance defined in Lemma C.7.

Proof. With the cross-entropy augmented reward $r_{\pi}(s,a)$ (Eq. (40)), the soft Bellman operator $\mathcal{T}^{\pi}$ in Eq. (17),

$$(\mathcal{T}^{\pi}Q)(s,a) = r(s,a) + \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[\alpha H\big(\pi(\cdot \mid s'), \tilde{\pi}(\cdot \mid s')\big) + \gamma \mathbb{E}_{a'\sim\pi(\cdot|s')}[Q(s',a')]\right],$$

is a γ-contraction in the infinite norm:

$$\|\mathcal{T}^{\pi}Q - \mathcal{T}^{\pi}Q'\|_{\infty} \leq \gamma \|Q - Q'\|_{\infty}.$$

Please note that $\mathcal{T}^{\tilde{\pi}}$ is the soft Bellman operator in MaxEnt-RL, which uses true entropy, while $\mathcal{T}^{\pi}$ is the Soft Bellman operator using cross-entropy, i.e.,

$$\left(\mathcal{T}^{\tilde{\pi}}Q\right)(s,a) = r(s,a) + \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[\alpha \mathcal{H}\big(\tilde{\pi}(\cdot \mid s')\big) + \gamma \mathbb{E}_{a'\sim\tilde{\pi}(\cdot|s')}[Q(s',a')]\right],$$

where the reward is augmented by the true entropy

$$r_{\tilde{\pi}}(s,a) = r(s,a) + \alpha \mathbb{E}_{a'\sim\tilde{\pi}(\cdot|s')}\left[\mathcal{H}\big(\tilde{\pi}(\cdot \mid s')\big)\right].$$

While the reward definitions differ, their discrepancy is controlled by the TV distance. Specifically, the difference is:

$$\begin{aligned} |r_{\pi}(s,a) - r_{\tilde{\pi}}(s,a)| &= \alpha \left|\mathbb{E}_{a'\sim\pi}[-\log\tilde{\pi}(a'|s)] - \mathbb{E}_{a'\sim\tilde{\pi}}[-\log\tilde{\pi}(a'|s)]\right| \\ &\leq 2\alpha \|\log\tilde{\pi}(\cdot|s)\|_{\infty} \cdot D_{\mathrm{TV}}(\pi(\cdot|s), \tilde{\pi}(\cdot|s)) \\ &\leq 2\alpha C_R \delta_{\mathrm{TV}}, \end{aligned} \tag{76}$$

where $C_R = \sup_{s,a} |\log\tilde{\pi}(a \mid s)|$ is a constant bounded by Assumption C.1.

Now, we decompose the difference between the optimal Q-functions:

$$\begin{aligned} \left\|Q^{\pi} - Q^{\tilde{\pi}}\right\|_{\infty} &= \left\|\mathcal{T}^{\pi}Q^{\pi} - \mathcal{T}^{\tilde{\pi}}Q^{\tilde{\pi}}\right\|_{\infty} \\ &\leq \left\|\mathcal{T}^{\pi}Q^{\pi} - \mathcal{T}^{\pi}Q^{\tilde{\pi}}\right\|_{\infty} + \left\|\mathcal{T}^{\pi}Q^{\tilde{\pi}} - \mathcal{T}^{\tilde{\pi}}Q^{\tilde{\pi}}\right\|_{\infty}. \end{aligned} \tag{77}$$

The first term is bounded by the contraction property of the Bellman operator:

$$\left\|\mathcal{T}^{\pi}Q^{\pi} - \mathcal{T}^{\pi}Q^{\tilde{\pi}}\right\|_{\infty} \leq \gamma \left\|Q^{\pi} - Q^{\tilde{\pi}}\right\|_{\infty}.$$

For the second term, we analyze the operator difference at any state-action pair (s,a):

$$\left|\left(\mathcal{T}^{\pi}Q^{\tilde{\pi}} - \mathcal{T}^{\tilde{\pi}}Q^{\tilde{\pi}}\right)(s,a)\right| \leq |r_{\pi}(s,a) - r_{\tilde{\pi}}(s,a)| + \gamma \left|\mathbb{E}_{s'}\left[\mathbb{E}_{a'\sim\pi}[Q^{\tilde{\pi}}(s',a')] - \mathbb{E}_{a'\sim\tilde{\pi}}[Q^{\tilde{\pi}}(s',a')]\right]\right|.$$

Using the reward bound derived above and the definition of total variation distance for the expectation term:

$$\begin{aligned} \|\mathcal{T}^{\pi}Q^{\tilde{\pi}} - \mathcal{T}^{\tilde{\pi}}Q^{\tilde{\pi}}\|_{\infty} &\leq 2\alpha C_R \delta_{\mathrm{TV}} + \gamma \cdot 2\|Q^{\tilde{\pi}}\|_{\infty}\delta_{\mathrm{TV}} \\ &= 2\delta_{\mathrm{TV}}\left(\alpha C_R + \gamma\|Q^{\tilde{\pi}}\|_{\infty}\right). \end{aligned} \tag{78}$$

Substituting these back into the triangle inequality (Eq. (77)) yields:

$$(1-\gamma)\|Q^{\pi} - Q^{\tilde{\pi}}\|_{\infty} \leq 2\delta_{\mathrm{TV}}\left(\alpha C_R + \gamma\|Q^{\tilde{\pi}}\|_{\infty}\right). \tag{79}$$

Since $\|Q^{\tilde{\pi}}\|_{\infty} \leq \frac{R_{\max}}{1-\gamma}$ and $\delta_{\mathrm{TV}} = \mathcal{O}(\sqrt{\mathrm{tr}(\Sigma)})$ completes the proof. □

C.2.2 SNIS Approach and Zeroth-Order Gradient

Fix an augmented state $\hat{s} = (s,z)$ and let the current local policy be Gaussian:

$$a = \mu + \delta, \qquad \mu = T_{\theta_k}(s,z), \qquad \delta \sim \mathcal{N}(0, \Sigma_k).$$

In the E-step of iteration k, the non-parametric target q_k takes the form

$$q_k(a \mid \hat{s}) \propto \hat{\pi}(a \mid \hat{s}; \theta_k) \cdot \exp\left(f_{\hat{s},k}(a)/\lambda\right), \qquad f_{\hat{s},k}(a) = Q^{\pi_k}(s,a) - \log\tilde{\pi}_{\theta_k}(a \mid s),$$

Moment matching via SNIS. The target mean in iteration k is given in Eq. (24). Using the reparameterization $a = \mu + \delta$, define $w(\delta) := \exp\left(f_{\hat{s},k}(\mu+\delta)/\lambda\right)$. Then

$$\mu^* = \frac{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[w(\delta)\,(\mu_k+\delta)\right]}{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[w(\delta)\right]}. \tag{80}$$

Hence the mean shift admits the normalized weighted-residual form

$$\Delta(\mu) := \mu^* - \mu = \frac{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[w(\delta)\,\delta\right]}{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[w(\delta)\right]}. \tag{81}$$

Connection to a zeroth-order gradient. In iteration k, we define the log-partition function $g(\mu)$:

$$g(\mu) := \log \mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[\exp\left(f(\mu+\delta)/\lambda\right)\right] = \log \mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}[w(\delta)] \tag{82}$$

Differentiating g with respect to μ gives

$$\nabla_\mu g(\mu) = \frac{\nabla_\mu \mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[w(\delta)\right]}{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[w(\delta)\right]} = \frac{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[\nabla_\mu w(\delta)\right]}{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[w(\delta)\right]} \tag{83}$$

Since $a = \mu + \delta$ implies

$$\frac{\partial a}{\partial \delta} = I,\ \frac{\partial a}{\partial \mu} = I \quad \longrightarrow \quad \nabla_\delta w(\delta) = \nabla_\mu w(\delta).$$

When we assume that $w(\delta)$ is sufficiently smooth and integrable with the boundary term equal to zero, we apply Stein's identity for $\delta \sim \mathcal{N}(0, \Sigma_k)$

$$\mathbb{E}_\delta\left[\delta\, w(\delta)\right] = \Sigma_k\, \mathbb{E}_\delta\left[\nabla_\delta w(\delta)\right]. \tag{84}$$

Combining Eq. (83) and Eq. (84) yields

$$\nabla_\mu g(\mu) = \frac{\mathbb{E}_\delta\left[\nabla_\delta w(\delta)\right]}{\mathbb{E}_\delta\left[w(\delta)\right]} = \Sigma_k^{-1}\,\frac{\mathbb{E}_\delta\left[\delta\, w(\delta)\right]}{\mathbb{E}_\delta\left[w(\delta)\right]} = \Sigma_k^{-1}\,\Delta(\mu). \tag{85}$$

Therefore, the direction to the target mean is the log-sum-exp estimate of zeroth-order gradient of $f_{\hat{s},k}(a)$:

$$\begin{aligned}\Delta(\mu) &= \Sigma_k \nabla_\mu g(\mu) \\ &= \Sigma_k \nabla_\mu \log \mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[\exp\left(f_{\hat{s},k}(\mu+\delta)/\lambda\right)\right].\end{aligned} \tag{86}$$

Remark. **From SAC's soft policy improvement to FLAG.** The soft policy improvement theorem of SAC [20] states that any policy π_{new} minimizing

$$D_{\text{KL}}\left(\pi(a \mid s)\,\middle\|\,\frac{\exp(Q^{\pi_{\text{old}}}/\alpha)}{Z}\right)$$

satisfies $Q^{\pi_{\text{new}}} \geq Q^{\pi_{\text{old}}}$. Importantly, the theorem is agnostic to how this KL minimizer is obtained. SAC restricts the policy class Π to a parametric Gaussian family and optimizes its parameters by gradient descent via reparameterization. In contrast, FLAG inherits the same improvement principle but solves the KL minimization through a non-parametric, action-level update (Figure 5).

Connection to the soft policy improvement theorem. Two ingredients align FLAG's update with the soft policy improvement theorem applied to the reference policy $\pi_{\text{old}} = \tilde{\pi}_{\theta_k}$.

- **Approximation of the soft objective.** Lemma C.8 shows that Q^π and $Q^{\tilde{\pi}_{\theta_k}}$ differ only by $\mathcal{O}(\sqrt{\text{tr}(\Sigma)})$. Therefore, FLAG's energy $f_{\hat{s},k}$ effectively realizes SAC's $\tilde{\pi}$-based soft objective.
- **Approximation of the action-gradient update.** Appendix C.2.2 shows that the moment-matching direction

  $$\mu_k^*(\hat{s}) - \mu_k = \Sigma_k \nabla_\mu g_k$$

  is a zeroth-order approximation of SAC's first-order action gradient

  $$\nabla_a\left(Q^{\tilde{\pi}}(s,a) - \alpha \log \tilde{\pi}_{\theta_k}(a \mid s)\right),$$

  via Stein's identity (Eq. (84)) and log-sum-exp smoothing (Eq. (86)).

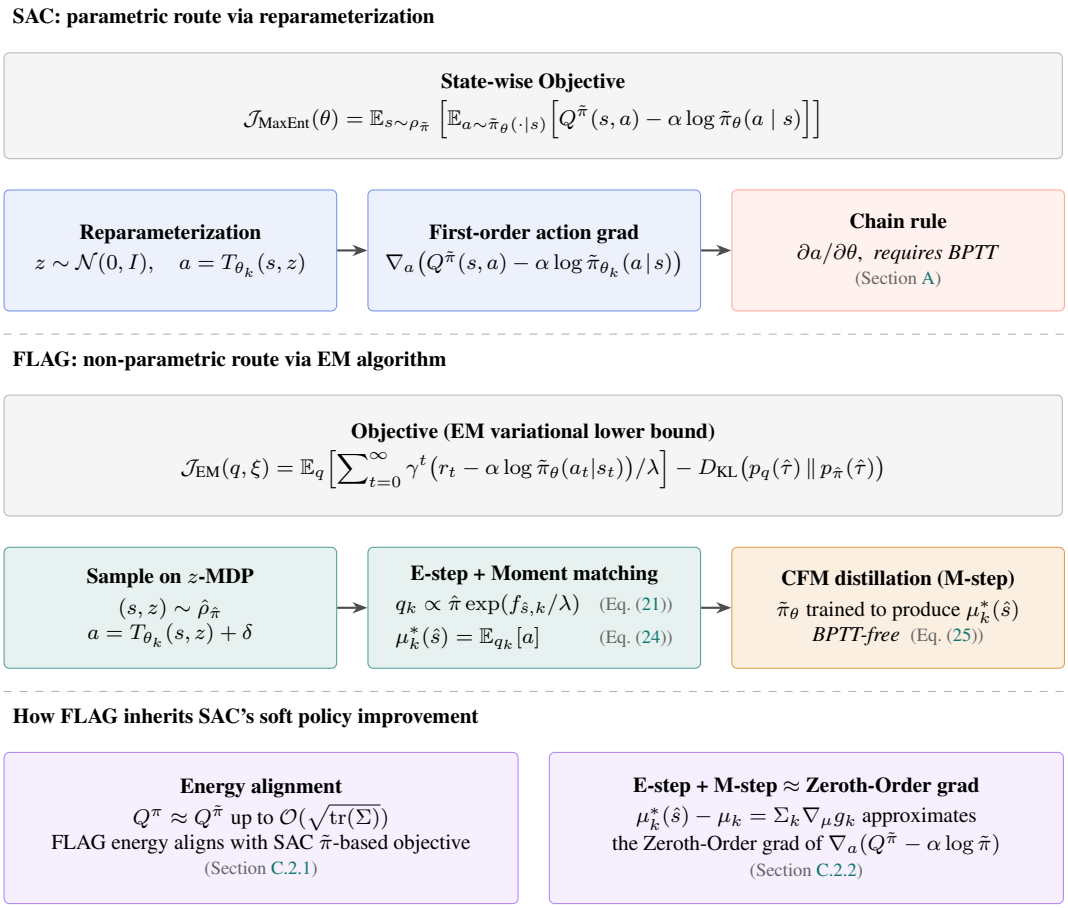


Figure 5: Comparison of SAC and FLAG at the k-th iteration. SAC realizes the soft policy improvement through reparameterization-based action gradients propagated to policy parameters via BPTT. FLAG instead solves an EM variational lower bound whose reward is regularized by cross-entropy. The two approaches are connected by (i) Q-function closeness between the composite policy π and base flow policy $\tilde{\pi}$ and (ii) the relation between FLAG's moment-matching update and a zeroth-order approximation of SAC's first-order action gradient.

Together, these two ingredients establish the SAC-based part of Theorem 4.5. The distinction between SAC's parametric route and FLAG's non-parametric route lies in the policy class Π and the corresponding optimization loss, not in the underlying soft policy improvement theorem.

What is distinctive about FLAG. The soft policy improvement theorem guarantees only that the target actions $\{\mu_k^*(\hat{s})\}_{\hat{s}}$ define an improved policy. It does not specify how a flow-based generative model should be trained to produce these actions. This is precisely where FLAG contributes: the M-step CFM distillation (Eq. (25)) provides a supervised, BPTT-free mechanism for training the flow policy to realize the improved target actions. In this way, FLAG provides a practical realization of SAC-style policy improvement for expressive flow-based policies.

C.3 Monotonic Improvement by Maximum a Posteriori Policy Optimisation

We now state a practically relevant monotonic-improvement result for FLAG in the latent-augmented MDP. We follow MPO [2] to prove the monotonic improvement in Eq. (19) and make some modifications since we are using cross-entropy augmented reward.

Setup and Notation. We work in a tabular z-MDP with finite augmented state space $\hat{\mathcal{S}}$ and a finite action grid $\mathcal{A}$. Each discrete action $a \in \mathcal{A}$ is identified with a bin $B_a \subset \mathbb{R}^{d_a}$ and a representative point $\bar{a} \in B_a$. The local Gaussian policy is defined on the underlying continuous action variable

$x \in \mathbb{R}^{d_a}$ and then projected onto the finite action grid:

$$\hat{\pi}_{\theta_k}(a \mid \hat{s}) := \int_{B_a} \varphi_{\Sigma_k}\big(x - \mu_{\theta_k}(\hat{s})\big)\, dx, \qquad \Sigma_k = \sigma_k^2 I. \tag{87}$$

All Gaussian mean-shift and covariance calculations below are performed in this underlying continuous action space. we identify each discrete action $a \in \mathcal{A}$ with its representative point $\bar{a}$. In particular, $\mu_k^*(\hat{s}) := \sum_{a\in\mathcal{A}} q_k(a \mid \hat{s})\, \bar{a}$. The induced discrete policy is obtained by bin projection. For notational simplicity, we write

$$\hat{\pi}_k(\cdot \mid \hat{s}) := \hat{\pi}_{\theta_k}(\cdot \mid \hat{s}), \qquad \tilde{\pi}_k(\cdot \mid s) := \tilde{\pi}_{\theta_k}(\cdot \mid s).$$

The reward in iteration k is given as

$$\hat{r}_k(\hat{s}, a) := r(s, a) - \alpha \log \tilde{\pi}_k(a \mid s). \tag{88}$$

For a reference policy $\hat{\pi}$ and a non-parametric policy q, define the λ-regularized reward

$$\hat{r}_{k,\lambda}^{\hat{\pi}_k, q}(\hat{s}, a) := \hat{r}_k(\hat{s}, a) - \lambda \log \frac{q(a \mid \hat{s})}{\hat{\pi}_k(a \mid \hat{s})}. \tag{89}$$

The corresponding Bellman operators are

$$(\hat{\mathcal{T}}_k^q \hat{V})(\hat{s}) = \mathbb{E}_{a\sim q(\cdot|\hat{s})}\Big[\hat{r}_k(\hat{s}, a) + \gamma \mathbb{E}_{\hat{s}'\sim\hat{p}(\cdot|\hat{s},a)}\big[\hat{V}(\hat{s}')\big]\Big], \tag{90}$$

$$(\hat{\mathcal{T}}_{k,\lambda}^{\hat{\pi}_k, q} \hat{V})(\hat{s}) = \mathbb{E}_{a\sim q(\cdot|\hat{s})}\Big[\hat{r}_{k,\lambda}^{\hat{\pi}_k, q}(\hat{s}, a) + \gamma \mathbb{E}_{\hat{s}'\sim\hat{p}(\cdot|\hat{s},a)}[\hat{V}(\hat{s}')]\Big]. \tag{91}$$

We also define the corresponding value functions

$$\hat{V}_k^q(\hat{s}) = \mathbb{E}_q\left[\sum_{t=0}^{\infty} \gamma^t \hat{r}_k(\hat{s}_t, a_t) \,\middle|\, \hat{s}_0 = \hat{s}\right], \tag{92}$$

$$\hat{V}_{k,\lambda}^{\hat{\pi}_k, q}(\hat{s}) = \mathbb{E}_q\left[\sum_{t=0}^{\infty} \gamma^t \left(\hat{r}_k(\hat{s}_t, a_t) - \lambda \log \frac{q(a_t \mid \hat{s}_t)}{\hat{\pi}_k(a_t \mid \hat{s}_t)}\right) \,\middle|\, \hat{s}_0 = \hat{s}\right]. \tag{93}$$

Using the energy $f_{\hat{s},k}(a)$ defined in Eq. (20), the target of constrained E-step in Eq. (41) is given by

$$q_k(\cdot \mid \hat{s}) = \arg\max_q \mathbb{E}_{a\sim q(\cdot|\hat{s})}\big[f_{\hat{s},k}(a)\big] - \lambda\big(D_{\mathrm{KL}}\big(q(\cdot \mid \hat{s}) \,\|\, \hat{\pi}_k(\cdot \mid \hat{s})\big) - \epsilon\big), \tag{94}$$

which admits the closed form

$$q_k(a \mid \hat{s}) = \frac{\hat{\pi}_k(a \mid \hat{s}) \exp\big(f_{\hat{s},k}(a)/\lambda\big)}{Z_k(\hat{s})}, \qquad Z_k(\hat{s}) = \sum_{a\in\mathcal{A}} \hat{\pi}_k(a \mid \hat{s}) \exp\big(f_{\hat{s},k}(a)/\lambda\big). \tag{95}$$

Finally, define the exact KL-projection objective

$$h(\hat{\pi}, q, \theta) := \mathbb{E}_{\hat{\pi},\hat{p}}\left[\sum_{t=0}^{\infty} \gamma^t D_{\mathrm{KL}}\big(q(\cdot \mid \hat{s}_t) \,\|\, \hat{\pi}(\cdot \mid \hat{s}_t; \theta)\big) \,\middle|\, \hat{s}_0, \hat{\pi}\right]. \tag{96}$$

Proposition C.9 (Bellman operator and Monotonicity)**.** *For every iteration k, the E-step target q_k satisfies*

$$\hat{\mathcal{T}}_{k,\lambda}^{\hat{\pi}_k, q_k} \hat{V}_k^{\hat{\pi}_k} \geq \hat{V}_k^{\hat{\pi}_k}. \tag{97}$$

Consequently, by the monotonicity of the Bellman operator, the regularized value function satisfies the following improvement for all states $\hat{s}$:

$$\hat{V}_{k,\lambda}^{\hat{\pi}_k, q_k} \geq \hat{V}_k^{\hat{\pi}_k}. \tag{98}$$

Proof. The proof closely follows the policy evaluation argument presented in MPO, adapted to our latent-augmented z-MDP and cross-entropy reward structure. By the definition of q_k in Eq. (95), for every $\hat{s} \in \hat{\mathcal{S}}$, we have $(\hat{\mathcal{T}}_{k,\lambda}^{\hat{\pi}_k, q_k} \hat{V}_k^{\hat{\pi}_k})(\hat{s}) \geq (\hat{\mathcal{T}}_{k,\lambda}^{\hat{\pi}_k, \hat{\pi}_k} \hat{V}_k^{\hat{\pi}_k})(\hat{s}) = \hat{V}_k^{\hat{\pi}_k}$. Since the z-MDP is finite and the augmented reward $\hat{r}_k$ is bounded (Assumption C.1), the operator $\hat{\mathcal{T}}_{k,\lambda}^{\hat{\pi}_k, q_k}$ remains monotone and γ-contractive in the ∞-norm. Therefore, repeatedly applying this operator and invoking the standard fixed-point theorem yields the final inequality $\hat{V}_{k,\lambda}^{\hat{\pi}_k, q_k} \geq \hat{V}_k^{\hat{\pi}_k}$. □

Objective decomposition. Next, we establish an MPO-style one-step improvement bound for the EM update in Section 4.3. In standard MPO [2], the reward is fixed across the E-step and M-step, so the EM update can be interpreted as a coordinate-ascent step on a KL-regularized variational objective. In FLAG, the augmented reward

$$\hat{r}_k(\hat{s}, a) = r(s, a) - \alpha \log \tilde{\pi}_{\theta_k}(a \mid s)$$

also depends on the base flow policy. Therefore, after the base flow is updated from θ_k to θ_{k+1}, the reward changes from $\hat{r}_k$ to $\hat{r}_{k+1}$. The proof below separates these two effects:

$$\Big(\underbrace{\textit{frozen-reward MPO improvement}}_{①} \Big) \quad + \quad \Big(\underbrace{\textit{cross-entropy reward drift}}_{②} \Big). \tag{99}$$

For brevity, define

$$\mathcal{J}_k := \mathcal{J}(q_k, \theta_k) = \mathbb{E}_{\hat{s}_0 \sim \hat{p}} \left[\hat{V}_{k,\lambda}^{\hat{\pi}_k, q_k}(\hat{s}_0) \right] \tag{100}$$

where q_k denotes the exact non-parametric E-step target at iteration k. For theoretical analysis, we first consider the ideal KL-projection update

$$\theta_{k+1}^{\mathrm{KL}} = \theta_k - \beta \nabla_\theta h(\hat{\pi}_k, q_k, \theta_k), \tag{101}$$

and denote the corresponding policy by $\hat{\pi}_{k+1}^{\mathrm{KL}}(\cdot \mid \hat{s}) := \hat{\pi}(\cdot \mid \hat{s}; \theta_{k+1}^{\mathrm{KL}})$. The practical CFM projection update is handled later in Lemma C.14.

Assumption C.10 (A Standard MPO regularity [2])**.** For each iteration k, the following hold:

1. The KL objective $h(\hat{\pi}_k, q_k, \theta_k)$ is L-smooth in a neighborhood of θ_k.
2. There exists $L_{\tilde{\pi}} > 0$ such that $\|\nabla_\theta \log \tilde{\pi}_\theta(a \mid s)\| \le L_{\tilde{\pi}}$ for all θ in a neighborhood of θ_k.
3. The E-step target and the KL-projected policy change only to first order in the M-step size:

$$\sup_{\hat{s}} \|q_{k+1}(\cdot \mid \hat{s}) - q_k(\cdot \mid \hat{s})\|_1 = \mathcal{O}(\beta), \qquad \sup_{\hat{s}} \|\hat{\pi}_{k+1}^{\mathrm{KL}}(\cdot \mid \hat{s}) - \hat{\pi}_k(\cdot \mid \hat{s})\|_1 = \mathcal{O}(\beta).$$

Assumption C.11 (Variance-scaled drift alignment)**.** Assume that, within iteration k, the local covariance is fixed with respect to θ, state-independent, and isotropic:

$$\Sigma_k = \sigma_k^2 I, \qquad \sigma_k^2 > 0.$$

Here Σ_k denotes the covariance matrix for notational convenience, while σ_k^2 denotes its scalar variance. Let the first-order reward-drift score be

$$\mathsf{G}_k^{\tilde{\pi}} := \mathbb{E}_{q_k} \left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \tilde{\pi}_{\theta_k}(a_t \mid s_t) \,\middle|\, \hat{s}_0 \right].$$

There exists a structural constant $C_\Sigma > 0$, independent of the step size β, such that

$$\left| \left\langle \mathsf{G}_k^{\tilde{\pi}}, \nabla_\theta h(\hat{\pi}_k, q_k, \theta_k) \right\rangle \right| \le C_\Sigma \sigma_k^2 \mathcal{G}_k. \tag{102}$$

Given Assumption C.10 then for any $0 \le \beta \le 1/L$, the descent lemma [33] gives

$$h(\hat{\pi}_k, q_k, \theta_{k+1}^{\mathrm{KL}}) \le h(\hat{\pi}_k, q_k, \theta_k) - \beta \mathcal{G}_k, \qquad \mathcal{G}_k := \tfrac{1}{2} \|\nabla_\theta h(\hat{\pi}_k, q_k, \theta_k)\|^2. \tag{103}$$

This decrease in the KL projection objective will produce the standard MPO-style improvement when the reward is fixed at $\hat{r}_k$. The additional difficulty is that the cross-entropy term $-\alpha \log \tilde{\pi}_\theta(a \mid s)$ also changes after the update. The following two assumptions isolate these two components.

Justification on Assumption C.11 While the following derivation relies on approximations, it establishes a structural bound on the drift term based on the zeroth-order approximation from Appendix C.2.2. This arises because the M-step parameter update $\Delta\theta_k$, which governs the drift, is shaped by the zeroth-order moment-matching direction. Following Section 4.1, we assume an isotropic covariance $\Sigma_k = \sigma_k^2 I$.

Let $\kappa_k(\hat{s};\theta) = D_{\mathrm{KL}}\big(q_k(\cdot \mid \hat{s}) \parallel \hat{\pi}(\cdot \mid \hat{s};\theta)\big)$ denote the statewise KL projection in the M-step. Using the zeroth-order approximation $\mu_k^*(\hat{s}) - \mu_k(\hat{s}) \approx \sigma_k^2 \nabla_\mu g_k(\mu_k(\hat{s}))$ from Eq. (86), the statewise M-step gradient with respect to θ is simplified to:

$$-\nabla_\theta \kappa_k(\hat{s};\theta_k) = J_\theta(\hat{s})^\top \sigma_k^{-2}\big(\mu_k^*(\hat{s}) - \mu_k(\hat{s})\big) \approx J_\theta(\hat{s})^\top v_k(\hat{s}), \tag{104}$$

where $J_\theta(\hat{s}) := \partial\mu_k(\hat{s})/\partial\theta$ and $v_k(\hat{s}) := \nabla_\mu g_k(\mu_k(\hat{s}))$. Consequently, the squared statewise gradient norm is independent of σ_k^{-2}:

$$\|\nabla_\theta \kappa_k(\hat{s};\theta_k)\|^2 \approx v_k(\hat{s})^\top J_\theta(\hat{s}) J_\theta(\hat{s})^\top v_k(\hat{s}). \tag{105}$$

In contrast, applying a first-order Taylor expansion to the smoothed energy yields the statewise base-flow drift direction:

$$\begin{aligned} J_\theta(\hat{s})^\top \Delta\mu &= J_\theta(\hat{s})^\top \Big(\mathbb{E}_{a\sim q_k(\cdot|\hat{s})}[a] - \mu_k(\hat{s})\Big) \\ &= J_\theta(\hat{s})^\top \frac{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[\delta \exp\left(f_{\hat{s},k}(\mu_k+\delta)/\lambda\right)\right]}{\mathbb{E}_{\delta\sim\mathcal{N}(0,\Sigma_k)}\left[\exp\left(f_{\hat{s},k}(\mu_k+\delta)/\lambda\right)\right]} \\ &= J_\theta(\hat{s})^\top \Sigma_k \nabla_\mu g_k(\mu_k(\hat{s})) \\ &= J_\theta(\hat{s})^\top \Sigma_k v_k(\hat{s}). \end{aligned} \tag{106}$$

To lift the statewise calculation to the global objective, we follow the standard MPO auxiliary-objective construction [2]. In the E-step, q_k is obtained as a statewise non-parametric improvement of the reference policy $\hat{\pi}_k$, while the subsequent projection is evaluated on states visited by $\hat{\pi}_k$. Therefore, we use the frozen discounted visitation measure $\hat{\rho}_k(\hat{s}) \equiv \hat{\rho}_{\hat{\pi}_k}(\hat{s})$ for the state weighting, and use $q_k(\cdot \mid \hat{s})$ only as the improved conditional action distribution at each visited state. Define the visitation-weighted mean-drift surrogate

$$\bar{\mathsf{G}}_k^{\tilde{\pi}} := \sum_{\hat{s}\in\hat{\mathcal{S}}} \hat{\rho}_k(\hat{s}) J_\theta(\hat{s})^\top \Delta\mu_k(\hat{s}). \tag{107}$$

This surrogate is the leading-order mean-shift approximation of the actual reward-drift score $\mathsf{G}_k^{\tilde{\pi}}$ in Assumption C.11.

Let $\mathbf{J}$ and $\mathbf{v}$ denote the corresponding state-stacked objects with weights $\sqrt{\hat{\rho}_k(\hat{s})}$. Then

$$U = \mathbf{J}^\top \mathbf{v}, \qquad D \approx \sigma_k^2 \mathbf{J}^\top \mathbf{v}, \qquad 2\mathcal{G}_k = \|U\|^2 = \mathbf{v}^\top \mathbf{J}\mathbf{J}^\top \mathbf{v}.$$

Hence the leading-order global alignment is

$$\big\langle \bar{\mathsf{G}}_k^{\tilde{\pi}}, -\nabla_\theta h(\hat{\pi}_k, q_k, \theta_k)\big\rangle \approx D^\top U = \sigma_k^2 \mathbf{v}^\top(\mathbf{J}\mathbf{J}^\top)\mathbf{v} = 2\sigma_k^2 \mathcal{G}_k. \tag{108}$$

Thus the leading-order calculation gives the variance-scaled drift alignment. The factor 2, the mismatch between the actual reward-drift score $\mathsf{G}_k^{\tilde{\pi}}$ and the mean-drift surrogate $\bar{\mathsf{G}}_k^{\tilde{\pi}}$, together with the zeroth-order, Taylor, and projection approximation errors, is absorbed into the constant C_Σ in Assumption C.11.

Proposition C.12 (Monotone Improvement in the z-MDP)**.** *Under Assumptions C.10 and C.11, there exists a constant $C > 0$, independent of β, such that the ideal KL update satisfies*

$$\mathcal{J}_{k+1}^{\mathrm{KL}} \geq \mathcal{J}_k + (\lambda - \alpha C_\Sigma \sigma_k^2)\beta \mathcal{G}_k - C\beta^2, \tag{109}$$

where

$$\mathcal{J}_{k+1}^{\mathrm{KL}} := \mathbb{E}_{\hat{s}_0\sim\hat{p}_0}\Big[\hat{V}_{k+1,\lambda}^{\hat{\pi}_{k+1}^{\mathrm{KL}}, q_{k+1}}(\hat{s}_0)\Big].$$

In particular, if

$$\lambda > \alpha C_\Sigma \sigma_k^2, \tag{110}$$

then, for sufficiently small β,

$$\mathcal{J}_{k+1}^{\mathrm{KL}} \geq \mathcal{J}_k. \tag{111}$$

Proof.

Step 1: We first write the decomposition point-wise for a fixed initial state $\hat{s}_0$. Taking expectation over $\hat{s}_0 \sim \hat{p}_0$ then gives the corresponding statement for $\mathcal{J}_{k+1}^{\mathrm{KL}} - \mathcal{J}_k$.

$$\begin{aligned}\mathcal{J}_{k+1}^{\mathrm{KL}} - \mathcal{J}_k &= \mathbb{E}_{\hat{s}_0\sim\hat{p}_0}\left[\hat{V}_{k+1,\lambda}^{\hat{\pi}_{k+1}^{\mathrm{KL}},q_{k+1}}(\hat{s}_0) - \hat{V}_{k,\lambda}^{\hat{\pi}_k,q_k}(\hat{s}_0)\right] \\ &= \mathbb{E}_{\hat{s}_0\sim\hat{p}_0}\Bigg[\underbrace{\left(\hat{V}_{k,\lambda}^{\hat{\pi}_{k+1}^{\mathrm{KL}},q_{k+1}}(\hat{s}_0) - \hat{V}_{k,\lambda}^{\hat{\pi}_k,q_k}(\hat{s}_0)\right)}_{①\text{ in Eq. (99)}} + \underbrace{\left(\hat{V}_{k+1,\lambda}^{\hat{\pi}_{k+1}^{\mathrm{KL}},q_{k+1}}(\hat{s}_0) - \hat{V}_{k,\lambda}^{\hat{\pi}_{k+1}^{\mathrm{KL}},q_{k+1}}(\hat{s}_0)\right)}_{②\text{ in Eq. (99)}}\Bigg].\end{aligned} \tag{112}$$

The first bracket freezes the reward at $\hat{r}_k$, so it has exactly the same form as the objective analyzed in MPO. The second bracket is the only new term in FLAG, and it appears because the reward changes when $\tilde{\pi}_k$ is updated to $\tilde{\pi}_{k+1}$.

Step 2: Fix the iteration k and freeze the augmented reward $\hat{r}_k$. In this case, the z-MDP objective has the same KL-regularized coordinate-ascent structure as MPO, with the substitutions $s \mapsto \hat{s}$, $\pi \mapsto \hat{\pi}$, and $r \mapsto \hat{r}_k$. Accordingly, define the frozen-reward auxiliary functional

$$\mathcal{H}_k(\hat{\pi}, q, \theta, \hat{\pi}') := \mathbb{E}_{\hat{\pi},\hat{p}}\left[\sum_{t=0}^{\infty}\gamma^t\left(\mathbb{E}_{a_t\sim q(\cdot|\hat{s}_t)}\left[\hat{Q}_k^{\hat{\pi}'}(\hat{s}_t,a_t)\right] - \lambda D_{\mathrm{KL}}\big(q(\cdot\mid\hat{s}_t)\,\|\,\hat{\pi}(\cdot\mid\hat{s}_t;\theta)\big)\right)\,\middle|\,\hat{s}_0\right]. \tag{113}$$

where $Q_k^{\hat{\pi}'}(\hat{s},a) := \hat{r}_k(\hat{s},a) + \gamma\mathbb{E}_{\hat{s}'\sim\hat{p}(\cdot|\hat{s},a)}\left[\hat{V}_k^{\hat{\pi}'}(\hat{s}')\right]$. By construction, $\mathcal{H}_k$ is exactly the MPO auxiliary objective specialized to the tabular z-MDP with frozen reward $\hat{r}_k$. Therefore, under Assumption C.10, the argument of [2, Appendix A.2] applies to Eq. (113) and yields

$$\hat{V}_{k,\lambda}^{\hat{\pi}_{k+1}^{\mathrm{KL}},q_{k+1}}(\hat{s}_0) - \hat{V}_{k,\lambda}^{\hat{\pi}_k,q_k}(\hat{s}_0) \geq \lambda\beta\mathcal{G}_k - C_1\beta^2, \tag{114}$$

for some constant $C_1 > 0$ independent of β. Equivalently, the frozen-reward part of the FLAG update inherits the same first-order improvement term $\lambda\beta\mathcal{G}_k$ as MPO, up to a second-order remainder $C_1\beta^2$.

Strictly, the MPO argument exploits the one-step optimality of the E-step target with respect to the frozen reward $\hat{r}_k$, while the actual q_{k+1} in FLAG is optimal with respect to the iteration-$(k+1)$ reward $\hat{r}_{k+1}$. The induced discrepancy is controlled by two first-order quantities: the Q-function drift and the E-step target drift. Since $\Delta\theta_k = \mathcal{O}(\beta)$, the reward drift $\hat{r}_{k+1} - \hat{r}_k$ is $\mathcal{O}(\beta)$, and combining this with Assumption C.10 gives

$$\left\|\hat{Q}_{k+1}^{\hat{\pi}_{k+1}} - \hat{Q}_k^{\hat{\pi}_{k+1}}\right\|_\infty = \mathcal{O}(\beta), \qquad \sup_{\hat{s}}\,\|q_{k+1}(\cdot\mid\hat{s}) - q_k(\cdot\mid\hat{s})\|_1 = \mathcal{O}(\beta).$$

Transferring the optimality of q_{k+1} from the iteration-$(k+1)$ objective to the frozen-reward objective then introduces a single cross-term of these two quantities,

$$\left(\mathbb{E}_{q_{k+1}} - \mathbb{E}_{q_k}\right)\left[\hat{Q}_k^{\hat{\pi}_{k+1}} - \hat{Q}_{k+1}^{\hat{\pi}_{k+1}}\right] = \mathcal{O}(\beta^2),$$

which is absorbed into the constant C_1 in Eq. (114).

Step 3: Due to the moving base flow policy, the reward-drift term in Eq. (112) is given by

$$\hat{V}_{k+1,\lambda}^{\hat{\pi}_{k+1}^{\mathrm{KL}},q_{k+1}}(\hat{s}_0) - \hat{V}_{k,\lambda}^{\hat{\pi}_{k+1}^{\mathrm{KL}},q_{k+1}}(\hat{s}_0) = \alpha\,\mathbb{E}_{q_{k+1}}\left[\sum_{t=0}^{\infty}\gamma^t\Big(\log\tilde{\pi}_k(a_t\mid s_t) - \log\tilde{\pi}_{k+1}^{\mathrm{KL}}(a_t\mid s_t)\Big)\,\middle|\,\hat{s}_0\right]. \tag{115}$$

We define this quantity as the reward-drift term $\Delta_k^{\mathrm{drift}}$.

Let the parameter update be $\Delta\theta_k := \theta_{k+1}^{\mathrm{KL}} - \theta_k = -\beta\nabla_\theta h(\hat{\pi}_k, q_k, \theta_k)$. By Taylor expanding $\log\tilde{\pi}_{\theta_{k+1}^{\mathrm{KL}}}(a\mid s)$ around θ_k, we obtain:

$$\begin{aligned}\Delta_k^{\mathrm{drift}} &= -\alpha\,\mathbb{E}_{q_{k+1}}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\tilde{\pi}_{\theta_k}(a_t\mid s_t)\,\middle|\,\hat{s}_0\right]^\top\Delta\theta_k + \mathcal{O}(\|\Delta\theta_k\|^2) \\ &= \alpha\beta\,\mathbb{E}_{q_{k+1}}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\tilde{\pi}_{\theta_k}(a_t\mid s_t)\,\middle|\,\hat{s}_0\right]^\top\nabla_\theta h(\hat{\pi}_k,q_k,\theta_k) + \mathcal{O}(\beta^2).\end{aligned} \tag{116}$$

By Assumption C.10, we have $\sup_{\hat{s}} \|q_{k+1}(\cdot \mid \hat{s}) - q_k(\cdot \mid \hat{s})\|_1 = \mathcal{O}(\beta)$. By Assumption C.10 and the boundedness of the base-flow score, replacing the discounted score expectation under q_{k+1} by that under q_k changes the first-order Taylor coefficient by $\mathcal{O}(\beta)$. Since this coefficient is multiplied by β, the resulting error is $\mathcal{O}(\beta^2)$.

$$\Delta_k^{\text{drift}} = \alpha\beta\big\langle \mathsf{G}_k^{\tilde{\pi}}, \nabla_\theta h(\hat{\pi}_k, q_k, \theta_k)\big\rangle + \mathcal{O}(\beta^2). \tag{117}$$

Plugging the Assumption C.11 into Eq. (117), the final bound on the reward-drift magnitude is

$$\big|\Delta_k^{\text{drift}}\big| \le \alpha C_\Sigma \sigma_k^2 \beta \mathcal{G}_k + C_d \beta^2 \tag{118}$$

for some constant $C_d > 0$ independent of β.

Substituting the reward-freeze improvement (① in Eq. (114)) and reward-drift bound (② in Eq. (118)) into the objective decomposition (① + ② in Eq. (112)), we conclude:

$$\begin{aligned}
\mathcal{J}_{k+1}^{\text{KL}} - \mathcal{J}_k &\ge \lambda\beta\mathcal{G}_k - C_1\beta^2 - \big|\Delta_k^{\text{drift}}\big| \\
&\ge \lambda\beta\mathcal{G}_k - C_1\beta^2 - \big(\alpha C_\Sigma \sigma_k^2 \beta \mathcal{G}_k + C_d\beta^2\big) \\
&= (\lambda - \alpha C_\Sigma \sigma_k^2)\beta\mathcal{G}_k - (C_1 + C_d)\beta^2.
\end{aligned} \tag{119}$$

Hence, if

$$\lambda > \alpha C_\Sigma \sigma_k^2, \tag{120}$$

then, for a sufficiently small step size β, the overall objective is nondecreasing:

$$\mathcal{J}_{k+1}^{\text{KL}} \ge \mathcal{J}_k.$$

□

Practical CFM update. Proposition C.12 analyzes the KL-based M-step. In practice, however, FLAG distills the resulting target into the base flow policy via the CFM loss (Eq. (25)). To bound the gap, we relate the practical update to the KL update through an excess projection error, rather than claiming that the CFM loss itself decreases $h(\hat{\pi}_k, q_k, \theta_k)$.

Assumption C.13 (Approximate realization of the ideal KL update by CFM)**.** There exists $\epsilon_k^{\text{proj}} \ge 0$ such that the practical CFM update satisfies

$$\mathcal{J}_{k+1}^{\text{CFM}} \ge \mathcal{J}_{k+1}^{\text{KL}} - \lambda\epsilon_k^{\text{proj}}, \tag{121}$$

where

$$\mathcal{J}_{k+1}^{\text{CFM}} := \mathbb{E}_{\hat{s}_0\sim\hat{p}_0}\left[\hat{V}_{k+1,\lambda}^{\hat{\pi}_{k+1}^{\text{CFM}}, q_{k+1}}(\hat{s}_0)\right], \qquad \mathcal{J}_{k+1}^{\text{KL}} := \mathbb{E}_{\hat{s}_0\sim\hat{p}_0}\left[\hat{V}_{k+1,\lambda}^{\hat{\pi}_{k+1}^{\text{KL}}, q_{k+1}}(\hat{s}_0)\right].$$

Lemma C.14 (Approximate monotone improvement under the practical CFM update)**.** *Under Assumptions C.10, C.11 and C.13, there exists a constant $C > 0$, independent of β, such that*

$$\mathcal{J}_{k+1}^{\text{CFM}} \ge \mathcal{J}_k + (\lambda - \alpha C_\Sigma \sigma_k^2)\beta\mathcal{G}_k - C\beta^2 - \lambda\epsilon_k^{\text{proj}}. \tag{122}$$

In particular, if

$$(\lambda - \alpha C_\Sigma \sigma_k^2)\beta\mathcal{G}_k \;\ge\; C\beta^2 + \lambda\epsilon_k^{\text{proj}}, \tag{123}$$

then

$$\mathcal{J}_{k+1}^{\text{CFM}} \ge \mathcal{J}_k.$$

Proof. By Assumption C.13,

$$\mathcal{J}_{k+1}^{\text{CFM}} \ge \mathcal{J}_{k+1}^{\text{KL}} - \lambda\epsilon_k^{\text{proj}}.$$

Applying Proposition C.12 to the ideal KL iterate yields

$$\mathcal{J}_{k+1}^{\text{KL}} \ge \mathcal{J}_k + (\lambda - \alpha C_\Sigma \sigma_k^2)\beta\mathcal{G}_k - C\beta^2.$$

Combining the two inequalities proves Eq. (122). The last statement follows immediately from Eq. (123). □

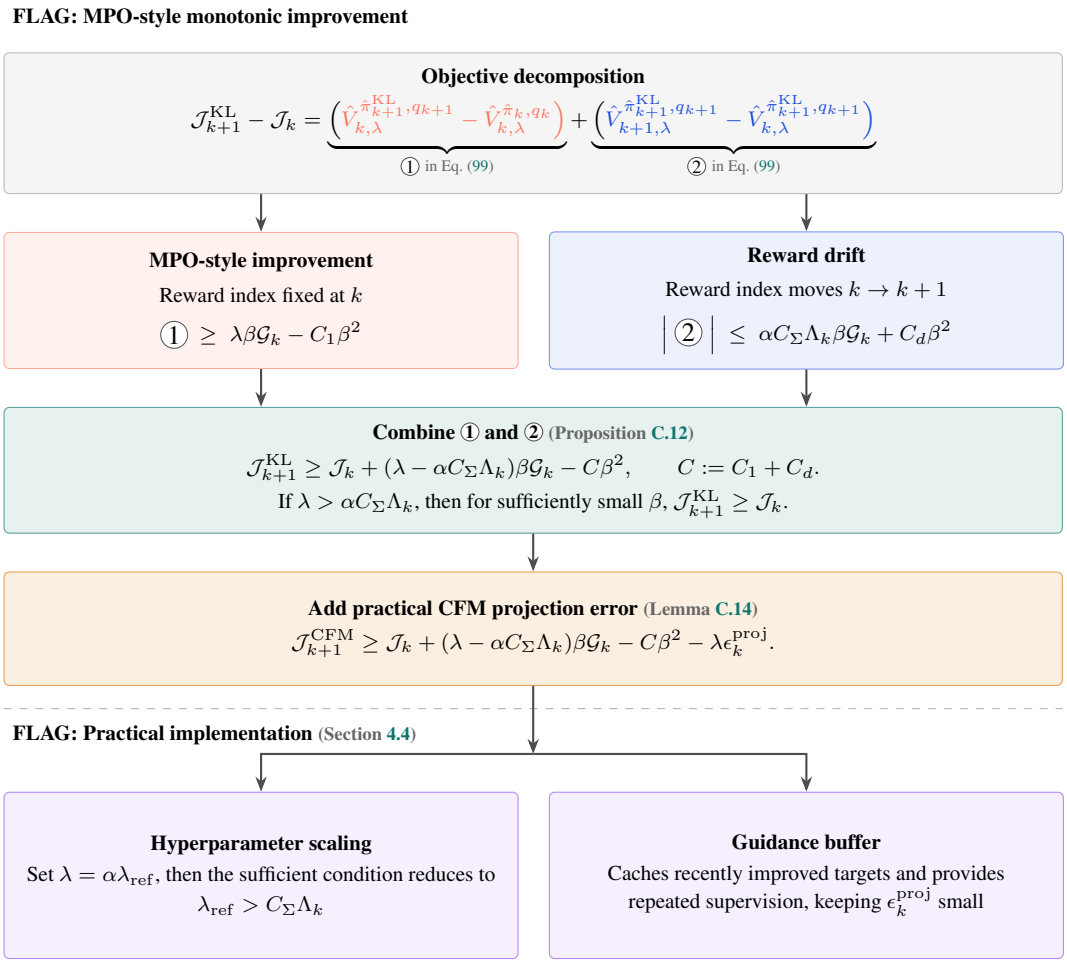


Figure 6: Proof roadmap for the MPO-style monotonic improvement of FLAG. The objective gap is decomposed into the standard MPO-style improvement term (① in Eq. (99)) and the FLAG-specific reward-drift term (② in Eq. (99)). Combining the two yields Proposition C.12, and adding the practical CFM projection error yields Lemma C.14. The bottom row shows two practical implementation choices that ensure the sufficient conditions: hyperparameter scaling $\lambda = \alpha\lambda_{\mathrm{ref}}$ and the guidance buffer for keeping $\epsilon^{\mathrm{proj}}_k$ small (Section 4.4).

Remark. **Remark: Practical implications of the bound.** The monotonic improvement guarantee in Proposition C.12 and Lemma C.14 carries two practical implications, both naturally satisfied by FLAG's design (see Figure 6). First, using the effective temperature $\lambda = \alpha\lambda_{\mathrm{ref}}$ (Section 4.4), the sufficient condition reduces to $\lambda_{\mathrm{ref}} > C_\Sigma\sigma^2_k$, which holds because covariance scheduling or clipping keeps σ^2_k small in practice. Second, the projection error $\epsilon^{\mathrm{proj}}_k$ is kept modest by the guidance buffer, which caches recently improved targets and provides dense, repeated supervision for the flow policy across off-policy updates.

D Implementation Details

D.1 Action scaling and SNIS details

We map pre–action samples u to bounded actions $a \in [-1, 1]^{|\mathcal{A}|}$ via an element–wise $\tanh$, where $|\mathcal{A}|$ is the cardinality of the action space. Let us denote the pre–action sampling distribution at iteration k as $r_k(u \mid \hat{s})$, which is a Gaussian centered on the flow policy output $\mu(\hat{s}; \theta_k)$ and has a scheduled covariance Σ_k. To apply change of variables to r_k, we need the determinant of the Jacobian (of the element–wise $\tanh$), $|\det J_{\tanh}(u)| = \prod_{i=1}^{|\mathcal{A}|}(1 - a_i^2)$ where $a = \tanh(u)$. Applying change

of variables leads to the following:

$$\hat{\pi}_k(a \mid \hat{s}) = \frac{r_k(u \mid \hat{s})}{|\det J_{\tanh}(u)|}, \qquad \log \hat{\pi}_k(a \mid \hat{s}) \;=\; \log r_k(u \mid \hat{s}) - \sum_{i=1}^{D} \log\bigl(1 - a_i^2\bigr). \tag{124}$$

We form the E-step target distribution following Eq. (21), only difference being the parameterization where we do not parameterize the covariance Σ_k, thus $\xi_k = \theta_k$. Pulling back to pre–action space via Eq. (124) yields

$$q_k(u \mid \hat{s}) = q_k(\tanh^{-1}(a) \mid \hat{s})\,|\det J_{\tanh}(u)| \propto r_k(u \mid \hat{s})\, \exp\left(\frac{f_{\hat{s},k}(\tanh(u))}{\alpha_k \lambda_{\text{ref}}}\right) \tag{125}$$

Using r_k as the proposal distribution, we define unnormalized importance weight w and subsequently self-normalized importance weight $\bar{w}$ with N pre–action samples. The moment matching in the pre–action space (u–space) is estimated by self-normalized importance sampling:

$$w(u) := \exp\left(\frac{f_{\hat{s},k}(\tanh(u))}{\alpha_k \lambda_{\text{ref}}}\right), \quad \bar{w}_i = \frac{w(u_i)}{\sum_{j=1}^{N} w(u_j)}, \quad \mu^*(\hat{s}) = \sum_{i=1}^{N} \bar{w}_i u_i. \tag{126}$$

D.2 Critic networks

We adapt the distributional Q-function from [5] to account for the entropy of the policy. The TD-target y in Eq. (17) is computed by

$$y = r(s,a) + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s')}\left[\sum_{i=0}^{b-1}\left(Q_{\min} + \left(\frac{Q_{\max} - Q_{\min}}{b-1}\right) i\right) \cdot p_i - \alpha \log \tilde{\pi}_\theta(a' \mid s')\right], \tag{127}$$

where b is the number of bins and p_i denotes the probability of the i-th bins which is the output of the critic network. After calculating the target we project it to discrete bins, $\hat{y} = \texttt{two\_hot}(y)$. The critic minimizes the following loss, which was proposed in [11].

$$\mathcal{L}_Q(\phi) = -\sum_{i=1}^{b} \hat{y} \log p_i - 0.005 \sum_{i=1}^{b} p_i \log p_i, \tag{128}$$

Following [31], we use the *mean* of the two TD-targets from two Q-function instead of *min*.

D.3 Variance Reduction in Hutchinson Trace Estimation

Rademacher Distribution. We sample ϵ from the Rademacher distribution, which typically yields a lower variance estimator [21, 3].

Common Random Numbers (CRN). When performing the estimation in Eq. 28, we evaluate the difference in log-probabilities between an action a and its perturbed neighbors $a + \delta$ to reduce the variance of the estimate compared to the case of estimating them individually. Specifically, the variance of the paired difference estimate can be minimized by employing the *Common Random Numbers* (CRN) technique [18]. This technique uses the exact same noise vector ϵ for both a and $a + \delta$, inducing a strong coupling between the two stochastic estimates.

Let $\hat{L}(a;\epsilon) = \epsilon^\top \nabla u(a) \epsilon$ denote the stochastic estimator for the trace term at a using a noise vector ϵ, where u is a vector field. The variance of the paired difference estimator is given by:

$$\mathrm{Var}(\hat{L}(a+\delta;\epsilon) - \hat{L}(a;\epsilon)) = \mathrm{Var}(\hat{L}(a+\delta;\epsilon)) + \mathrm{Var}(\hat{L}(a;\epsilon)) - 2\,\mathrm{Cov}(\hat{L}(a+\delta;\epsilon), \hat{L}(a;\epsilon)). \tag{129}$$

Assuming that the Jacobian $\nabla u(a)$ varies smoothly when δ is small, $\hat{L}(a+\delta;\epsilon)$ and $\hat{L}(a;\epsilon)$ are positively correlated under the shared ϵ. Consequently, the large covariance term cancels a substantial portion of the individual variances, yielding a lower-variance estimate of the difference.

Application to Weight Calculation. To leverage the variance-reducing property of CRN, we formulate the importance sampling weights in terms of *differences*. We hereafter omit the iteration index k for brevity. Ideally, the updated action μ^* is computed as the weighted average of candidate actions using weights $w(a) = \exp(f_{\hat{s}}(a))$. We observe that the SNIS ratio is invariant to a constant baseline shift. Dividing $w(a)$ by $\exp(f_{\hat{s}}(\mu))$, we define the baseline subtracted weight $\tilde{w}(a) := \exp(f_{\hat{s}}(a) - f_{\hat{s}}(\mu))$:

$$\mu^* = \sum_{i=1}^{N} \frac{\exp\left(f_{\hat{s}}(a_i)\right) \cdot a_i}{\sum_{j=1}^{N} \exp\left(f_{\hat{s}}(a_j)\right)} = \sum_{i=1}^{N} \frac{\tilde{w}(a_i) \cdot a_i}{\sum_{j=1}^{N} \tilde{w}(a_j)}. \tag{130}$$

$\exp(f_{\hat{s}}(a) - f_{\hat{s}}(\mu))$ contains a log-probability difference term $\log \tilde{\pi}(a \mid s) - \log \tilde{\pi}(\mu \mid s)$, allowing us to directly apply the CRN technique described above.

D.4 Learning the Covariance Network

Covariance Σ_ψ in Eq. (8) is learnable via second moment matching, leveraging the target mean estimates from D.3. We assume diagonal covariance, where a neural network predicts $\sigma = f_\psi(s) \in \mathbb{R}^{|\mathcal{A}|}$ and equivalently $\Sigma = \sigma^2 I \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$. The target variance of the d-th action as the weighted second moment is given by

$$(\sigma_d^*)^2 \approx \sum_{i=1}^{N} \frac{\bar{w}(a_i)\,(a_{i,d} - \mu_d^*)^2}{\sum_{j=1}^{N} \bar{w}(a_j)}. \tag{131}$$

Preventing Deterministic Collapse via Entropy Regularization. When the temperature α becomes small, the normalized weights $\bar{w}(a_i)$ can collapse toward a one-hot distribution, causing $\sigma_i \to 0$. This drives the covariance network toward a nearly deterministic policy and eliminates local exploration. To counteract this effect, we add an explicit entropy-preserving force based on the entropy of the local Gaussian policy. For $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_{|\mathcal{A}|}^2)$, the entropy of the local policy is $\mathcal{H}_{\text{local}} = \sum_{i=1}^{|\mathcal{A}|} \log \sigma_i + \text{const.}$, whose derivative w.r.t. σ_i is $1/\sigma_i$. Combining this entropy regularization term with the term induced by moment matching yields the following update:

$$\nabla_{\sigma_i} \mathcal{J}_{\text{total}} \approx \underbrace{\frac{(\sigma_i^*)^2 - \sigma_i^2}{\sigma_i^3}}_{\text{second moment matching}} + \underbrace{\beta \frac{1}{\sigma_i}}_{\text{entropy regularization}}. \tag{132}$$

The equilibrium yields $\sigma_i^2 = \frac{(\sigma_i^*)^2}{1-\beta}$ where larger $\beta \in [0, 1)$ enlarges the target variance relative to the pure moment matching and helps prevent variance collapse.

Log-Space Target Matching. In practice, the covariance network predicts the log-standard deviation $\varsigma_i = \log \sigma_i$. Applying the chain rule gives

$$\frac{\partial \mathcal{J}_{\text{total}}}{\partial \varsigma_i} = \frac{\partial \mathcal{J}_{\text{total}}}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial \varsigma_i} = \frac{(\sigma_i^*)^2}{\sigma_i^2} - 1 + \beta. \tag{133}$$

Eq. (133) identifies the desired equilibrium, but directly optimizing this asymmetric gradient can be numerically unstable in bounded parameterizations of ς_i. Instead, we construct a surrogate loss whose minimizer matches the same equilibrium in log-space. From the equilibrium $\sigma_i^2 = \frac{(\sigma_i^*)^2}{1-\beta}$, the corresponding target log standard deviation is $\varsigma_{\text{target},i} = \frac{1}{2} \log\left((\sigma_i^*)^2\right) - \frac{1}{2} \log(1 - \beta)$. To avoid the singularity at $(\sigma_i^*)^2 = 0$, we introduce a minimum variance floor $\sigma_{\min}^2$ and define the stabilized exploitation target covariance $\hat{\Sigma}_i = \max((\sigma_i^*)^2, \sigma_{\min}^2)$. We then compute the bounded target

$$\varsigma_{\text{target},i} = \mathrm{clip}\left(\frac{1}{2} \log \hat{\Sigma}_i - \frac{1}{2} \log(1 - \beta), \varsigma_{\min}, \varsigma_{\max}\right). \tag{134}$$

Finally, we update the covariance network using the symmetric log-space MSE objective

$$\mathcal{L}(\psi) = \mathbb{E}_{s \sim \mathcal{D}} \left[\frac{1}{D} \sum_{i=1}^{D} \left(\varsigma_i(s) - \mathtt{sg}(\varsigma_{\text{target},i}(s)) \right)^2 \right], \tag{135}$$

where $\mathtt{sg}(\cdot)$ is a stop-gradient operator.

Algorithm 1 FLAG: Flow Policy MaxEnt-RL by Latent Augmented Guidance

Initialize critic networks Q_{ϕ_1}, Q_{ϕ_2}, and vector field network u_θ with random parameters ϕ_1, ϕ_2, θ.
Initialize environment replay buffer $\mathcal{D}_{\text{ENV}}$ and small guidance buffer $\mathcal{D}_{\text{FLAG}}$
for $k = 1$ to M **do**
  if $k \,\%\, \text{UTD} = 0$ **then**
    Sample $a \sim \pi(a \mid s; \theta_k)$
    Step environment: $s' \sim p(s' \mid s, a)$
    Store (s, a, r, s') in $\mathcal{D}_{\text{ENV}}$
  end if
  Sample B transitions $(s, a, s', r) \sim \mathcal{D}$
  for each update step **do**
    Sample B transitions (s, a, r, s') from $\mathcal{D}$
    Sample $a' \sim \pi(a' \mid s'; \theta_k)$
    Compute the log probability of $\tilde{\pi}$ (Eq. (28))
    Update critics: $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}(\phi)$ (Eq. (128))
    if $K \,\%$ POLICY DELAY **then**
      Sample N latent vectors $z_1, \dots z_N \sim p_z(z)$ and transport to actions $\{T_{\theta_k}(s, z_i)\}_{i=1}^N$
      Compute the log probability of $\tilde{\pi}$ (Eq. 28)
      Estimate $f_{\hat{s},k}(a_i)$ (Eq. (20)) and normalize with α
      Estimate μ^* (Eq. (22)) and Update base flow policy: $\theta_{k+1} \leftarrow \theta_k - \eta_k \nabla_\theta \mathcal{L}(\theta)$ (Eq. (25))
      Store (s, z, μ^*) in $\mathcal{D}_{\text{FLAG}}$
      Update $\alpha_{k+1} \leftarrow \alpha_k - \eta_\alpha \mathcal{J}(\alpha)$ (Eq. (27))
    end if
    Sample B augmented-state–action pairs $(s, z, \mu) \sim \mathcal{D}_{\text{FLAG}}$
    Update base flow policy: $\theta_{k+1} \leftarrow \theta_k - \eta_\theta \nabla_\theta \mathcal{L}(\theta)$ (Eq. (25))
  end for
end for

E Baseline

E.1 Baselines in Section 5.1

MaxEntDP[6] utilizes the Q-weighted Noise Estimation (QNE) method to approximate the exponential of the target distribution of MaxEnt-RL. In the original paper, They employ $N = 500$ action samples to ensure a low-variance and accurate training target for the noise prediction network, which is expensive. The policy naturally balances local and global update characteristics because the diffusion time step t controls the noise schedule α_t, which directly determines the scale of the sampled action area a_t during the training process. Our implementation follows their official Github repository.

DPMD[7] utilizes **Reweighted Score Matching (RSM)** to optimize diffusion policies without requiring direct samples from the optimal policy. The algorithm weights the score matching objective with the exponential of the Q-function, $\exp(Q(s, a)/\lambda)$. To ensure efficient exploitation and manage the inherent randomness of the diffusion process, DPMD adopts batch action sampling—a form of soft rejection sampling—where the action $a = \arg\max_i Q(s, a_{(i)})$ is selected from numerous generated candidates ($P = 32$) for environment interaction, whereas $N = 1$ action is sampled in the policy update. The inference time is more expensive than FLAG, because we use "policy delay" for computational efficiency in policy update, while the batch action sampling cannot. To match the sampling budget with FLAG, we modify the original DPMD so that the algorithm does not leverage batch action sampling heuristics ($P = 1$) and use multiple action samples ($N \leq 64$) during policy update. Our implementation follows their official Github repository.

QVPO[8] derives its policy update by leveraging a lower bound on the policy gradient objective. Specifically, rather than directly maximizing $\mathbb{E}_{a\sim\pi_\theta}[Q(s, a)]$ via gradient ascent, QVPO shows that this objective admits a tractable lower bound expressible as a Q-weighted score matching loss,

[6] https://github.com/diffusionyes/MaxEntDP
[7] https://github.com/mahaitongdae/diffusion_policy_online_rl
[8] https://github.com/wadx2019/qvpo

enabling the use of a diffusion model as the policy. Concretely, at each update step, $N = 64$ candidate actions are drawn from the current diffusion policy for each state, and the top-k ($k = 1$) actions by the minimum of two Q-networks are retained as positive targets. Only candidates whose Q-value exceeds a threshold ($Q > 1.0$) receive a nonzero weight, acting as a quality filter. The diffusion policy is then trained via the weighted denoising score matching objective. For exploration, QVPO augments the training batch with $N_{\text{neg}} = 10$ actions per positive sample drawn uniformly at random from $\mathcal{U}(-1, 1)$; these negative samples are assigned weights $w_i = \alpha \cdot Q_{\text{pos}}$ where α follows a linear decay schedule from 0.02 to 0.002 over 2×10^5 steps, encouraging broad exploration early in training. Our implementation follows their official Github repository.

For CrossQ update and distributional critic architecture we follows DIME official Github.

E.2 Baselines in Section 5.2

DIME [11]. For MuJoCo tasks, we run DIME[9] using the implementation released in the official GitHub repository. For DMC Dog and MyoSuite tasks, we directly use the results reported in the original paper, together with the reported CrossQ and QSM baselines.

QSM [38]. For MuJoCo tasks, we run QSM[10] using the official implementation. For a fair comparison, we align its hyperparameters with those used in DIME whenever applicable.

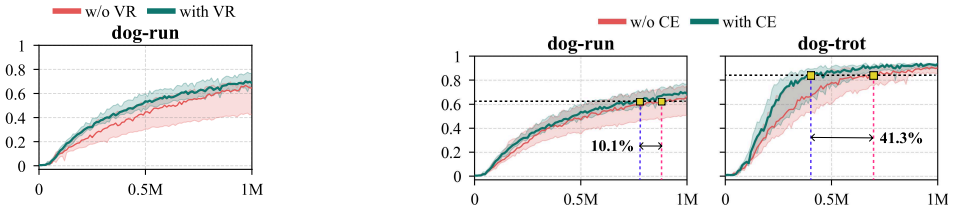
DACERv2 [48] and DIPO [49]. Our implementations of DACERv2[11] and DIPO[12] are based on their official GitHub repositories

Following DIME, we additionally incorporate the CrossQ update for the critic and use a distributional critic architecture.

FlowRL [26]. Our implementation of FlowRL[13] follows the official GitHub repository. FlowRL uses layer normalization [4] in the critic architecture and employs a three-layer MLP with hidden dimensions $[512, 512, 512]$, which is architecturally different from the wider CrossQ critic with hidden dimensions $[2048, 2048]$. Therefore, we do not apply the CrossQ modification to FlowRL and instead retain the original critic architecture.

All hyperparameters are listed in Table 8.

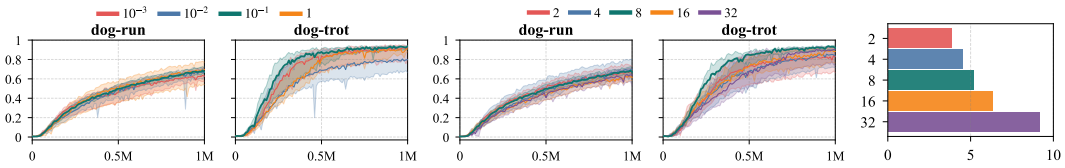
[9] https://github.com/ALRhub/DIME
[10] https://github.com/escontra/score_matching_rl
[11] https://github.com/happy-yan/DACER-Diffusion-with-Online-RL
[12] https://github.com/BellmanTimeHut/DIPO
[13] https://github.com/bytedance/FlowRL



(a) Variance Reduction on Trace Estimation

(b) Effect of Cross-Entropy

Figure 7: **Variance Reduction and Cross-Entropy Ablations.** *(a)* Training stability with and without the variance reduction technique for Hutchinson's trace estimator. *(b)* We compare FLAG against a variant trained without the cross-entropy term. The horizontal dashed line represents 90% of peak performance, with intersection points marked by ■.



(a) Sensitivity to λ_{ref}

(b) Effect of ODE solver steps

(c) Mean runtime (H)

Figure 8: **Ablation and Efficiency Analysis.** *(a)* Performance curves across different values of the KL-divergence constraint parameter λ_{ref}. *(b)* Aggregate performance as a function of the number of integration steps used during flow inference. *(c)* Runtime comparison.

F Additional Ablation Studies

F.1 Variance Reduction in Hutchinson's Trace Estimator

To efficiently estimate the log-likelihood of the flow policy, we employ Hutchinson's trace estimator. To mitigate the stochasticity of this approximation, we adopt a variance reduction technique that shares the random tangent vectors across the action samples, as detailed in Appendix D.3. We ablate this design choice to quantify its impact on training stability (Figure 7a). The results demonstrate that this variance reduction strategy not only accelerates learning but also significantly reduces the variance across random seeds, leading to more robust convergence.

F.2 Cross-entropy.

Although MaxEnt RL promotes systematic exploration, exact entropy computation for expressive policies is infeasible. We circumvent this by introducing a cross-entropy term as a tractable surrogate. To verify this design, we conduct an ablation study comparing our approach with a standard formulation that lacks the MaxEnt component. We omit the log-probability from the soft Q-function update (Eq. 16) and the weight calculation (Eq. 21). The model with cross-entropy reaches the 90% of peak performance approximately 10.1% and 41.3% faster in respective tasks compared to the baseline in Figure 7b. This shows that the inclusion of cross-entropy regularizes the policy by preventing overfitting to early value overestimation, as evidenced by the reduced variance across random seeds.

F.3 Sensitivity to λ_{ref}

We examine the sensitivity of FLAG to the hyperparameter λ_{ref}, reporting performance on the DMC dog-run and dog-trot tasks in Figure 8a. This parameter controls the allowable divergence between the updated local policy and the reference policy; intuitively, a higher λ_{ref} permits more aggressive updates, while a lower value enforces a tighter trust region. As illustrated, performance is sensitive to this choice. Extreme values—either overly restrictive or excessively unstable—degrade performance. For our main experiments, we found $\lambda_{\text{ref}} = 10$ to yield the most consistent results across tasks.

Table 4: **Learned covariance variant.** The learned covariance variant does not consistently outperform FLAG and shows sensitivity to the hyperparameters β and $\log \sigma_{\text{final}}$, making it less practical despite its closer alignment with the theoretical results.

	$\log \sigma_{\text{final}}$	β			
		0.0	**0.1**	**0.5**	**0.9**
Dog-Run	1e-3	684 [645, 738]	646 [613, 693]	672 [635, 735]	651 [571, 724]
	5e-4	484 [344, 672]	597 [562, 704]	645 [562, 698]	618 [550, 716]
Dog-Trot	1e-3	916 [885, 939]	910 [664, 929]	931 [909, 945]	914 [765, 933]
	5e-4	891 [767, 922]	876 [746, 930]	923 [901, 947]	899 [753, 930]

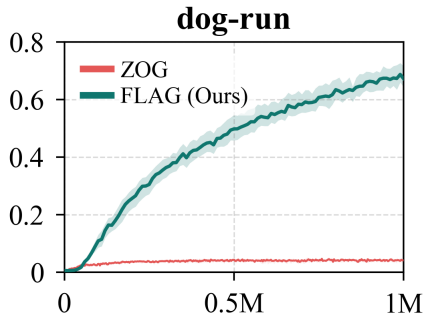


Figure 9: **Zeroth-order gradient variant.** The zeroth-order gradient variant of FLAG performs worse than FLAG, showing that our SNIS-based moment matching outperforms and is more stable than the zeroth-order gradient variant.

F.4 Influence of ODE Solver Steps

We analyze the trade-off between inference precision and computational cost by varying the number of ODE solver steps used for the flow policy. The results are summarized in Figure 8b. This result shows that the performance is relatively robust to the number of ODE solver steps. Since the training duration increases proportionally with the number of steps (Figure 8c), we fixed the solver steps to 8 for all experiments to achieve computational efficiency.

F.5 Covariance Learning and Connections to Zeroth-Order Gradient Methods

FLAG's moment-matching update admits a zeroth-order gradient interpretation. The mean shift $\mu_k^*(\hat{s}) - \mu_k$ from the E-step is equivalent to $\Sigma_k \nabla_\mu g_k(\mu_k)$, where $g_k(\mu) = \log \mathbb{E}_{\delta \sim \mathcal{N}(0,\Sigma_k)}[\exp(f_{\hat{s},k}(\mu+\delta)/\lambda)]$ is the log-sum-exp smoothing of the energy function, connecting FLAG's update to a zeroth-order estimate of the true action gradient $\nabla_a(Q^{\tilde{\pi}}(s,a) - \alpha \log \tilde{\pi}(a|s))$. From this perspective, Σ_k plays a dual role: it directly scales the zeroth-order gradient estimate, and simultaneously determines the width of the local Gaussian proposal $\hat{\pi}(a|\hat{s};\theta) = \mathcal{N}(a; T_\theta(s,z), \Sigma_k)$, controlling the search region around each anchor action.

Theoretically, a smaller Σ_k is preferable as it reduces both the Q-function discrepancy between π and $\tilde{\pi}$ and the smoothing bias of the gradient estimate, motivating the use of the learned covariance variant described in Section D.4. However, as shown in Table 4, the learned covariance variant does not consistently outperform FLAG and exhibits sensitivity to the choice of β and $\log \sigma_{\text{final}}$, making it less practical despite its closer theoretical alignment. Simple linear annealing therefore remains our design choice, as it maintains a well-calibrated gradient scale and a sufficiently wide search region without additional hyperparameter sensitivity.

One might naturally ask whether FLAG's moment-matching update could be replaced by a direct zeroth-order gradient update, which estimates the action gradient from samples without the EM

framework. However, as shown in Figure 9, this zeroth-order gradient variant performs substantially worse than FLAG, failing to discover high-rewarding action sequences. This highlights a key advantage of FLAG's SNIS-based moment matching over naïve sample-based gradient estimation: by reweighting samples according to the E-step target distribution rather than following a raw gradient direction, FLAG produces more informed and stable policy updates, while also benefiting from the broader search region of the annealing schedule as an implicit exploratory mechanism.

F.6 GPU Memory Allocation Comparison with DIME

DIME computes updates through direct gradient flow, which necessitates BPTT and leads to memory costs that grow rapidly with model size. FLAG avoids this bottleneck entirely by replacing gradient-based updates with a supervision loss, decoupling memory consumption from the temporal depth of the computation graph. As shown in Table 5, BPTT causes memory usage to escalate sharply as the actor grows larger, whereas FLAG scales substantially more efficiently. At 1B parameters, FLAG reduces memory consumption by over 43% relative to DIME, and savings reach as high as 75% at the 10M scale. These results demonstrate that a supervision-based update scheme is a practical necessity for scaling actor networks into the billion-parameter regime.

Table 5: GPU memory usage across actor parameter scales

	Actor Parameters				
Algorithm	**100K**	**1M**	**10M**	**100M**	**1B**
DIME (MB)	146	329	1133	1771	14220
FLAG (MB)	110	117	283	1381	8094
Memory Reduction (%)	24.66	64.44	75.02	22.02	43.08

G Experiment Details

We conduct all experiments using JAX [9]. For FlowRL [26], we follow their implementation with PyTorch [35].

G.1 Multigoal Environment

We conduct a small-scale experiment in the `MultiGoalEnv`, adapted from [19], to qualitatively assess whether SDAC, DPMD, QVPO, MaxEntDP, and FLAG can recover a multimodal action distribution from a shared value function. An optimal Q-function $Q(s, a)$ is first computed via dynamic programming on a discrete grid and then frozen across all methods as a common scoring oracle, isolating policy extraction from representation learning. Each method optimizes a diffusion or flow matching policy to approximate the Boltzmann target $\pi^*(a \mid s) \propto \exp(Q(s, a)/\tau)$ with $\tau = 0.1$, and we visualize the learned action distributions at two probe states, $(0, 0)$ and $(3, 3)$, via kernel density estimation. By fixing the Q-function, this setup directly measures each method's ability to capture the target distribution as a function of the number of importance samples $N \in \{2, 8, 32\}$ per gradient step, without confounding factors from Q-learning or environment interaction.

G.2 Comparison with Target Matching Methods

G.2.1 Figure 3.

Scores are averaged over four tasks per benchmark: HalfCheetah, Ant, Walker2d, and Humanoid for MuJoCo; Dog-run, Dog-trot, Dog-walk, and Dog-stand for DMC Dog; and reach-hard, obj-hold-hard,

Table 6: Environment state and action space dimensions

	Ant-v5	HalfCheetah-v5	Humanoid-v5	Walker2d-v5	DMC dog	MyoSuite
$\dim(\mathcal{S})$	105	17	348	17	223	93
$\dim(\mathcal{A})$	8	6	17	6	38	39

Table 7: Hyperparameters of experiments in Figure 3

	SDAC	**DPMD**	**QVPO**	**MaxEntDP**	**FLAG**
H_{target}	$-0.9\dim(\mathcal{A})$	$-0.9\dim(\mathcal{A})$	N/A	N/A	$-\dim(\mathcal{A})$
Temp. Learn. Rate	7e-3	7e-3	N/A	N/A	1e-3
Critic Learn. Rate	3e-4	3e-4	3e-4	3e-4	3e-4
Actor/Score Learn. Rate	3e-4 $\rightarrow$ 3e-5	3e-4 $\rightarrow$ 3e-5	3e-4 $\rightarrow$ 3e-5	3e-4	3e-4
Diffusion Steps	20	20	20	20	8
Discount	0.99	0.99	0.99	0.99	0.99
Batch size	256	256	256	256	256
Buffer size	1e6	1e6	1e6	1e6	1e6
Critic Hidden Depth	2	2	3	2	2
Critic Hidden Size	2048	2048	2048	2048	2048
Critic CrossQ	True	True	True	True	True
Critic LayerNorm	False	False	False	False	False
Dist. Critic Bins	101	101	101	101	101
Actor/Score Depth	3	3	3	3	3
Actor/Score Size	256	256	256	256	256
Update-to-data Ratio	2	2	2	2	2
Policy delay	3	3	3	3	3
Exploration Steps	10000	10000	10000	10000	10000

Table 8: Hyperparameters of experiments in Table 2

	DIME	**FlowRL**	**DACERv2**	**QSM**	**DIPO**	**CrossQ**	**FLAG**
H_{target}	$4\dim(\mathcal{A})$	N/A	$-0.9\dim(\mathcal{A})$	N/A	N/A	$-\dim(\mathcal{A})$	$-\dim(\mathcal{A})$
Temp. Learn. Rate	1e-3	N/A	3e-2	N/A	N/A	3e-4	1e-3
Critic Learn. Rate	3e-4	3e-4	1e-4	3e-4	3e-4	7e-4	3e-4
Actor/Score Learn. Rate	3e-4	3e-4	1e-4	3e-4	3e-4	7e-4	3e-4
Diffusion Steps	16	1	20	15	100	N/A	8
Discount	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	1e6	1e6	1e6	1e6	1e6	1e6	1e6
Critic hidden depth	2	3	2	2	2	2	2
Critic hidden size	2048	512	2048	2048	2048	2048	2048
Critic use CrossQ	True	False	True	True	True	True	True
Critic use LayerNorm	False	True	False	False	False	False	False
Num. Bin/Quantiles	101	N/A	2	N/A	101	N/A	101
Actor/Score depth	3	2	3	3	4	3	3
Actor/Score size	256	512	256	256	256	256	256
Update-to-data ratio	2	1	1	1	2	2	2
Policy delay	3	1	1	1	3	3	3
Exploration Steps	5000	10000	10000	10000	10000	5000	10000
Prior Distr.	$\mathcal{N}(0, 2.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	N/A	$\mathcal{N}(0, 1)$
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Score-Q align. factor	N/A	N/A	N/A	50	N/A	N/A	N/A

Table 9: List of FLAG-specific hyperparameters

α_{init}	λ_{ref}	$\log \sigma_{\text{init}}$ / $\log \sigma_{\text{min}}$	$\log \sigma$ Warmup / Decay Steps
0.01	10	-2 / -3	$2e5$ / $8e5$
Supervision Warmup / Ramp Steps	**Flow Buffer Size**	**Number of z given s in update**	**# of samples for Variance Reduction**
$1e5$ / $1e5$	10240	1	1

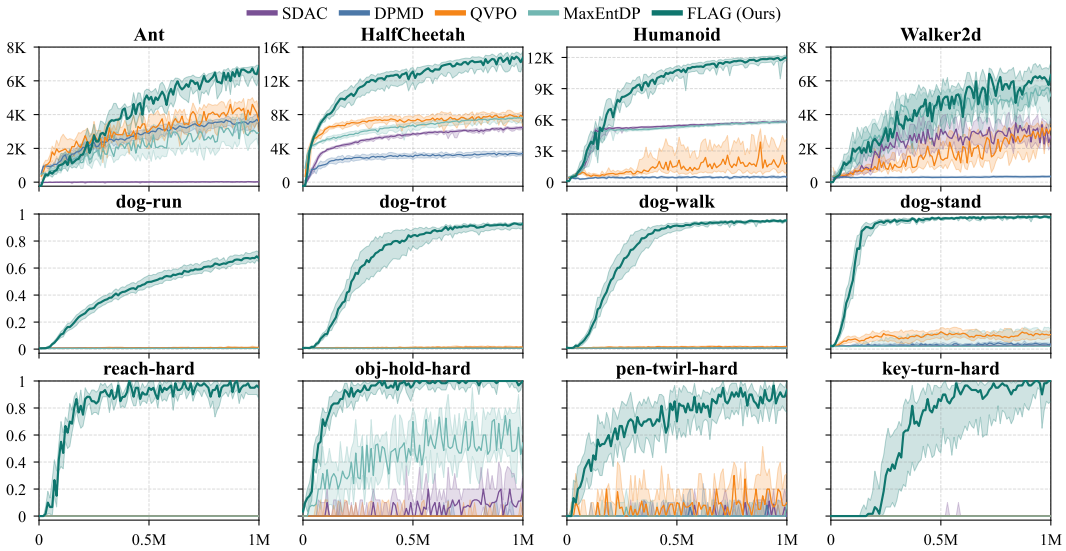


Figure 10: Full learning curves of Figure 3 (P=1)

key-turn-hard, and pen-twirl-hard for MyoSuite. Environment state and action space dimensions are summarized in Table 6. All methods are evaluated at 1M environment steps using 5 evaluation episodes, with IQM returns and confidence intervals computed over 10 seeds for $P = 1$ and 5 seeds for $P = 32$ due to the computational burden. GPU hours are measured on a single NVIDIA L40S GPU. Hyperparameters are reported in Table 7, and FLAG-specific hyperparameters shared across all experiments are listed in Table 9. Full learning curves corresponding to Figure 3 are provided in Figures 10 and 11.

G.2.2 Figure 4

We report performance at 1M steps using 5 evaluation episodes, with mean and standard deviation over 10 seeds for $P = 1$ and 5 seeds for $P = 32$, consistent with Figure 3. Hyperparameters follow those used for Figure 3 and Table 2, except for CrossQ usage and critic depth and hidden size. Specifically, we use $[256, 256, 256]$ critic architectures without Layer Normalization or Batch Renormalization. Please see Table 10 for detailed results.

G.3 Comparisons with Diffusion and Flow-based Algorithms

Table 2. We report performance at 1M environment interaction steps using 5 evaluation episodes with 10 random seeds per algorithm. Hyperparameters are reported in Table 8, and additional learning curves for the four MuJoCo tasks are provided in Figure 12.

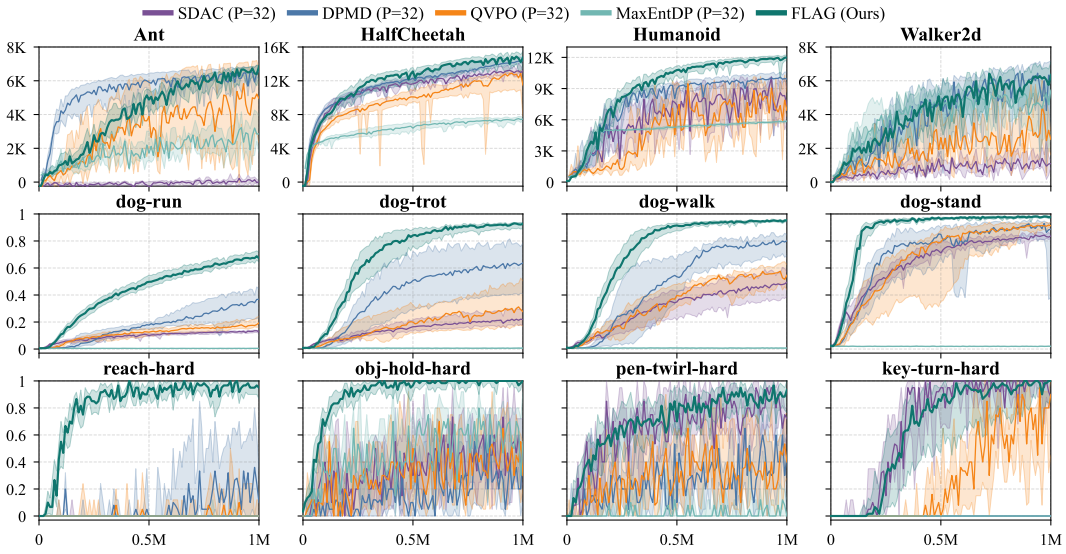


Figure 11: Full learning curves of Figure 3 (P=32)

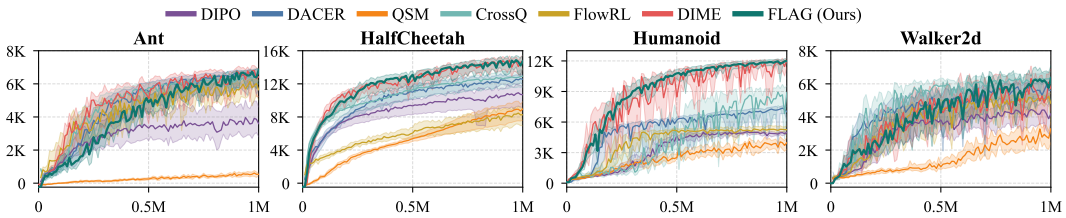


Figure 12: Learning curves other diffusion & flow-matching algorithms in 4 MuJoCo tasks

Table 10: **DMC Dog performance without CrossQ in Section 5.1.** Δ_F denotes the relative performance drop from FLAG, defined as $(1 - \text{Ret}_{\text{Alg}}/\text{Ret}_{\text{FLAG}}) \times 100$. † denotes the baselines' policy update relevant default hyperparameters used in the original papers. The highest and second scores are highlighted.

P	N	Alg.	Dog-trot		Dog-run	
			Return (1k)↑	Δ_F ↓	Return (1k)↑	Δ_F ↓
1	64	SDAC	0.008 ± 0.001	0.984	0.006 ± 0.000	0.980
	64	DPMD	0.013 ± 0.008	0.974	0.006 ± 0.002	0.980
	64	QVPO	0.020 ± 0.022	0.960	0.039 ± 0.036	0.872
	500	MaxEntDP†	0.007 ± 0.001	0.986	0.005 ± 0.001	0.984
32	64	SDAC†	0.019 ± 0.023	0.962	0.007 ± 0.002	0.977
32	1	DPMD†	0.117 ± 0.125	0.766	0.023 ± 0.016	0.924
4	64	QVPO†	0.077 ± 0.082	0.846	0.045 ± 0.054	0.852
–	–	DIME†	0.088 ± 0.073	0.824	0.025 ± 0.033	0.918
–	–	DACERv2†	0.107 ± 0.043	0.786	0.065 ± 0.059	0.786
1	8	FLAG†(ours)	0.500 ± 0.026	0.0	0.304 ± 0.027	0.0

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 three claims stated in the introduction are each substantiated in the paper.

   - The latent-augmented MDP and the proxy MaxEnt-RL objective with theoretical consistency are introduced in Section 4.2 and proved in Appendix B.
   - The EM-based FLAG algorithm with a monotonic improvement guarantee is presented in Section 4.3 and proved in Appendix C.
   - Empirical superiority over global IS baselines and state-of-the-art performance are demonstrated across DMC Dog, MyoSuite, and MuJoCo benchmarks in Section 5 (Figures 3 and 4, Tables 1 to 3).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 explicitly identifies the main limitation: the specific combination of a base flow policy and a local Gaussian head is one instantiation of the proposed framework, and a more principled construction via ODE-to-SDE conversion [15] is left for future work. In addition, the monotonic improvement guarantee (Theorem 4.5) relies on the approximation assumptions stated from C.10 to C.13, whose violation in practice is discussed in the surrounding remarks.

   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: All standing assumptions (from C.1 to C.4) and iteration-specific assumptions (from C.10 to C.13) are explicitly stated in Section C. Full proofs are provided for every formal result: Corollaries 4.2 and 4.4 in Appendix B, Proposition 4.3 in Appendix C.1, the SAC-perspective connection in Appendix C.2, and the MPO-style monotonic improvement (C.12, C.14) in Appendix C.3. Proof sketches are embedded in the main text (Section 4.3) and a visual roadmap is provided in Figure 6.

   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 complete training procedure is given in Algorithm 1 and Appendix D. All FLAG-specific hyperparameters are listed in Table 9, and full hyperparameter tables for the comparison experiments are provided in Tables 7 and 8. Baseline implementations reference official GitHub repositories (Appendix E.1), and any deviations from the originals are described in detail. Environment dimensions are reported in Table 6, and evaluation protocols (number of seeds, evaluation episodes, metric definitions) are specified in Section 5 and Appendix G.

   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: A codebase to reproduce the results introduced in this paper is presented with the paper. All environments used (DMC, MyoSuite, MuJoCo) are publicly available benchmarks, and all baseline implementations are linked to their official repositories in Appendix E.1.

   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: Optimizer (Adam), learning rates, batch sizes, buffer sizes, network architectures, discount factors, update-to-data ratios, and all other training details are fully reported in Tables 7 to 9. FLAG-specific design choices (covariance scheduling, guidance buffer size, effective temperature) are explained in Section 4.4 and further ablated in Section 5.3 and Appendix F.

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: All results are reported using the Interquartile Mean (IQM) with 95% confidence intervals computed over 10 random seeds (5 for the $P{=}32$ condition, as noted in Section 5.1). Each policy is evaluated with 5 evaluation episodes per seed. Regarding the $\pm$ notations in Tables 1 and 2, we directly adopt the notation from [23].

   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: Figure 3 reports wall-clock GPU hours for a single 1M-step training run on an NVIDIA L40S GPU for all compared methods, serving as the primary compute reference. Figure 8c provides a runtime bar chart comparing FLAG across different numbers of ODE solver steps, and notes that training duration increases proportionally with solver steps. The chosen solver step count of 8 is justified by this efficiency analysis.

   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: This work studies continuous control in simulated physics environments and involves no human subjects, personal data, or sensitive applications. No aspects of the research raise ethical concerns under the NeurIPS Code of Ethics.

   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 is foundational research on reinforcement learning algorithms evaluated exclusively on standard simulated locomotion benchmarks. There is no direct path to deployment or to harmful applications arising from improvements in simulation-based continuous control. Potential downstream benefits—such as more capable robotic controllers—are generic to the field and do not require specific discussion here.

    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 releases neither pre-trained models nor datasets. The proposed algorithm operates on simulated locomotion tasks and does not pose misuse risks that would necessitate safeguards.

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: All baseline algorithms (SDAC, DPMD, QVPO, MaxEntDP, DIME, FlowRL, DACERv2, DIPO, QSM) are citepd with their original papers and their official GitHub repositories are linked in Appendix E.1. The benchmark environments DMC [46], MyoSuite [10], and MuJoCo [43] are all citepd. The CrossQ critic implementation is similarly citepd [7].

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should citep 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: [N/A]

Justification: The paper does not introduce new datasets, benchmarks, or pre-trained model checkpoints as standalone assets.

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 involves no crowdsourcing and no research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper involves no human subjects and therefore requires no IRB approval or equivalent review.

    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: No LLMs are used as a component of the proposed method, the experimental setup, or the evaluation procedure. LLMs were not used in the core methodology or scientific contributions of this work.

    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.