Aligning Flow Map Policies with Optimal Q-Guidance

Christos Ziakas[1]* **Alessandra Russo**[1] **Avishek Joey Bose**[1, 2]

[1]Imperial College London [2]Mila

Abstract

Generative policies based on expressive model classes, such as diffusion and flow matching, are well-suited to complex control problems with highly multimodal action distributions. Their expressivity, however, comes at a significant inference cost: generating each action typically requires simulating many steps of the generative process, compounding latency across sequential decision-making rollouts. We introduce *flow map policies*, a novel class of generative policies designed for fast action generation by learning to take arbitrary-size jumps—including one-step jumps—across the generative dynamics of existing flow-based policies. We instantiate flow map policies for offline-to-online reinforcement learning (RL) and formulate online adaptation as a trust-region optimization problem that improves the critic's Q-value while remaining close to the offline policy. We theoretically derive FLOW MAP Q-GUIDANCE (FMQ), a principled closed-form learning target that exactly solves the linearized critic-guided trust-region problem for adapting offline flow map policies. We further introduce Q-GUIDED BEAM SEARCH (QGBS), a stochastic flow-map sampler that combines renoising with beam search to enable iterative inference-time refinement. Across 12 challenging robotic manipulation and locomotion tasks from OGBench and RoboMimic, FMQ achieves state-of-the-art performance in offline-to-online RL, outperforming the previous one-step policy MVP by a relative improvement of 21.3% on the average success rate.

1 Introduction

The supreme promise of offline reinforcement learning (RL) is that effective policies can be bootstrapped in a scalable data-driven manner without costly environment interaction [Levine et al., 2020]. This scaling philosophy is central to modern *data-driven* reinforcement learning [Kumar, 2019, Fu et al., 2020] that utilizes ever-growing diverse offline datasets [Collaboration et al., 2023], and now powers learning policies in high-impact applications from dialogue [Jaques et al., 2019] to robotic navigation [Kahn et al., 2018]. Indeed, imitating the highly multi-modal action distribution of expert behavior policies in such complex control problems necessitates the use of expressive policy classes that go beyond restrictive unimodal Gaussian actors [Zhu et al., 2023, Wang et al., 2023].

Generative policies based on dynamic mass transport, such as diffusion models [Sohl-Dickstein et al., 2015, Song et al., 2021] and flow-matching [Liu et al., 2023, Lipman et al., 2023, Albergo et al., 2023], provide a compelling alternative to Gaussian policies as they learn to map a simple base distribution into a rich state-conditioned action distribution [Chi et al., 2025]. The expressivity gains of generative policies make them particularly favorable for offline and offline-to-online RL [Fujimoto and Gu, 2021, Tarasov et al., 2023], where the policy must first model diverse behaviors from a static dataset and then improve through interaction. However, the price of expressive generative policies is computationally expensive inference-time simulation. More precisely, generating actions requires

*Correspondence to c.ziakas24@imperial.ac.uk

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

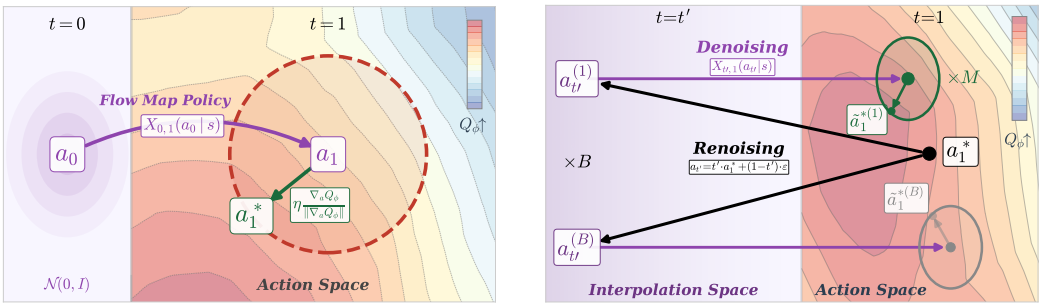


Figure 1: (Left) **FMQ**: one-step flow map policy transports noise a_0 to action a_1; then, trust-region projection displaces action a_1 to a_1^* that maximizes Q-value. (Right): **QGBS** (M=1, B=2): renoising corrupts a_1^* into B intermediate states $a_{t'}$, which the flow map policy then denoises to generate B candidate actions; candidates are updated via the trust-region displacement to maximize Q_ϕ, and the highest-valued M actions are selected.

numerically integrating dynamics from noise to action and is executed at every environment step [Yang et al., 2023]—inhibiting deployment in online and real-world settings [Zhan et al., 2024, 2025].

Addressing the inference latency of generative policies is of critical interest in order to extract maximal utility from generative policies, with several recent efforts attempting to learn one-step offline policies [Park et al., 2025b, Zhan et al., 2026]. However, in the context of offline-to-online RL, one-step generation remains insufficient as the policy must improve beyond the offline behavioral prior. In addition, current one-step generative policies commonly rely on heuristic-based generate-and-select procedures, such as best-of-N sampling, to bias actions toward high critic Q-values. Critically, this heuristic approach offsets the computational advantage of one-step policies by requiring many policy and critic evaluations per decision, while simultaneously not guaranteeing—for any finite N—*optimal local improvement* of the action sampled under the critic.

Present work. In this work, we study *flow map policies*, which learn the unique two-time jump operator associated with the probability flow ODE of diffusion and flow-matching policies [Boffi et al., 2024, 2025]. Flow map policies provide a unifying view of existing one-step policies for offline-to-online RL, e.g., mean velocity policies [Zhan et al., 2026], and make the established Lagrangian, Eulerian, and Progressive self-distillation objectives available for policy learning. In stark contrast to prior work, for principled online adaptation of one-step flow map policies, we formulate a trust-region optimization problem and derive a closed-form update that is optimal for its linearized subproblem and aligns the action distribution with Q-value guidance. This yields our contribution FLOW MAP Q-GUIDANCE (FMQ), which constructs a novel self-bootstrapped learning target—depicted in fig. 1—as the projected action-gradient of the critic, and removes distillation networks and best-of-N selection from the actor-training target.

We additionally introduce a complementary inference-time search procedure at evaluation, Q-GUIDED BEAM SEARCH (QGBS) that combines stochastic sampling through renoising candidate samples to an intermediate state with Q-guided beam search around the trust region. Importantly, QGBS produces diverse refinements around high Q-value actions without costly ODE simulations or additional learning during online adaptation. We summarize our core contributions as follows:

1. **Flow map policies**. We present flow map policies as a unifying framework for learning one-step policies as two-time jump operators for flow-based generative actors.
2. **Algorithms**. We introduce FMQ, which efficiently adapts flow map actors using the closed-form solution of a linearized trust-region problem (Theorem 3.2), with a nonlinear improvement guarantee under critic smoothness. We further introduce QGBS, a stochastic inference-time refinement algorithm that combines flow map renoising, beam selection, and trust-region Q-guidance.
3. **State-of-the-Art Performance:** Across 12 manipulation and locomotion tasks from OGBench and RoboMimic, FMQ outperforms our reproduction of the prior SOTA one-step baseline by a relative 21.3% while converging on average approximately 2.77× faster during online adaptation.

2 Background and Preliminaries

Offline-to-Online RL. We consider a Markov Decision Process (MDP) [Sutton and Barto, 1998] defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$, where $\mathcal{S} \subseteq \mathbb{R}^n$, $\mathcal{A} \subseteq \mathbb{R}^d$ denote continuous state-action spaces, $r(s, a)$ the reward function, $P(s'|s, a)$ the transition probability distribution, and $\gamma \in [0, 1)$ the discount factor. The objective of reinforcement learning is to train a policy $\pi(a|s)$ that maximizes the expected cumulative discounted return, $J(\pi) = \mathbb{E}_\pi\big[\sum_{\tau=0}^{\infty} \gamma^\tau r(s^\tau, a^\tau)\big]$, where τ denotes the timestep. Offline-to-online RL is a two-stage learning framework consisting of offline pre-training followed by online fine-tuning. In the offline pre-training phase, a behavioral prior policy is trained on a static dataset $\mathcal{D} = \{(s, a, r, s')\}$, providing an initialization. Subsequently, during online fine-tuning, the policy directly interacts with the environment. A popular approach is actor-critic methods, which employ an actor $\pi(a \mid s)$ and a critic $Q_\theta(s, a)$ that approximates the expected discounted return under policy π: $Q^\pi(s, a) = \mathbb{E}_{\pi,P}\left[\sum_{i=0}^{\infty} \gamma^i r(s^i, a^i) \;\middle|\; s^0 = s, a^0 = a\right]$.

Flow matching polices. A generative policy $a_1 \sim \pi_1(\cdot|s)$ conditioned on state s, can be formulated as transport plan which pushes forward an easy to sample reference measure $p_0(a_0) \in \mathcal{P}(\mathbb{R}^d)$ to a desired measure of (optimal) target action $p_1(a_1) := p^*_{\text{target}}(a^*) \in \mathcal{P}(\mathbb{R}^d)$. The subscripts are indicative of a notion of time where the process evolves a (pseudo)-action from the prior at time $t = 0$, i.e. $a_0 \sim p_0$, to an action that follows the target distribution $a_1 \sim p_1(a_1)$ at time $t = 1$. We highlight that the time t associated with the transport dynamics is distinguished from τ, which is associated with the MDP. Formally, a *flow-policy* is a one-parameter diffeomorphism, conditioned on a state s, $\psi_t(\cdot|s) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ that is the solution to the following ordinary differential equation:

$$\frac{d}{dt}\psi_t(a_t \mid s) = v_t\left(\psi_t(a_t \mid s)\right), \quad \psi_0(a_0) = a_0, \tag{1}$$

with initial conditions $\psi_0(a_0) = a_0$. Furthermore, $v_t : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ is a time-dependent (instantaneous) velocity field. In effect, the thesis of the generative policy problem is to learn a policy that pushes forward the base measure as follows $\pi_1(\cdot|s) := p_1 = [\psi_1(\cdot|s)]_\#(p_0)$. We highlight that ψ_1 produces a *deterministic* action while $\pi_1(\cdot|s)$ is induced *distribution* over actions at $t = 1$ by the flow-policy. To build the flow-policy, we can associate it with a conditional probability path $p_t(\cdot|z) : [0, 1] \times \mathcal{P}(\mathbb{R}^d) \to \mathcal{P}(\mathbb{R}^d)$ which is a time-indexed interpolation in probability space between two distributions p_0 and p_1. In its simplest form, the conditioning variable can be taken to the endpoints $z := (a_0, a_1)$, and a particle level interpolation that is simply a convex combination of the endpoints can be employed, i.e., $a_t = (1-t)a_0 + ta_1$. We say ψ_t generates p_t if it pushes forward p_0 to p_1 by following the ODE in eq. (1). To learn the flow policy, it is easier to regress against a known target conditional velocity $v^*_t(a_t|z, s)$ field that generates p_t. With access to such a v^*_t, learning can proceed using the conditional flow-matching loss [Tong et al., 2023, Albergo et al., 2023, Liu et al., 2023, Lipman et al., 2023, Peluchetti, 2023], which is a simple simulation-free regression objective: $\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,q(z),p_t(a_t|z,s)}\|v_t(a_t \mid s) - v^*_t(a_t \mid z, s)\|_2^2 = \mathbb{E}_{t,q(z),p_t(a_t|z,s)}\|v_t(a_t \mid s) - (a_1 - a_0)\|_2^2$, where $q(z)$ is a coupling over the states—e.g., independent coupling $q(z) = p_0(a_0)p_1(a_1)$—and in the last equality we substitute $v^*(a_t \mid z, s) = a_1 - a_0$ with its analytic linear speed target velocity.

3 Flow Map Q-Guidance

Generative policies operate over an inner-time axis, i.e., ODE simulation time, that is distinct from the evolution of the MDP. Consequently, for every state along the trajectory s^τ, a corresponding action a_1^τ must be generated through numerical simulation of the flow-policy ODE in eq. (1)—necessitating large amounts of function evaluations of the policy network. We next address this computational inefficiency through learning the flow-map, which dramatically speeds up simulation by taking large jumps along the ODE trajectory. We organize the remainder of this section as follows: in §3.1 we introduce *flow-map policies* and apply them to offline RL in §3.2. In §3.3 we rigorously design an efficient online adaptation update using a trust-region based on the critic's Q-function. Finally, in §3.4 we introduce our stochastic sampling approach to refine generated actions at inference.

3.1 Flow Map Policies

For high-fidelity action generation using flow-matching policies, it remains critical to simulate the infinitesimal dynamics of the parametrized velocity field in eq. (1). Instead of solving the ODE, we can parametrize and learn the unique two-time operator associated with the flow-matching policy.

Definition 3.1 (Flow-Map Policy). *Let $X_{r,t} : [0,1]^2 \times \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}^d$ be a flow map that evolves the action dynamics between any $(r,t) \in [0,1]$, conditioned on the MDP state $s \in \mathcal{S}$ governed by eq. (1), and satisfying the jump condition $X_{r,t}(a_r|s) = a_t$. The flow-map policy is then the distribution induced by this map evaluated at time $t = 1$, where $\pi(a|s) = [X_{r,1}]_{\#} p_r(a_r|s)$.*

To parametrize the underlying flow map that induces this policy, we leverage the *average action velocity* $u_{r,t} : [0,1]^2 \times \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}^d$, between the two time points r, t with the condition $r \leq t$:

$$X_{r,t}(a_r \mid s) = a_r + (t-r)u_{r,t}(a_r \mid s), \quad u_{r,t}(a_r \mid s) = \frac{1}{t-r}\int_r^t v_\tau(a_\tau \mid s)d\tau. \tag{2}$$

We note that, using eq. (2), we take jumps of size $t-r$ along the ODE trajectory. Furthermore, evaluating this flow map at the boundaries $r = 0$ and $t = 1$ yields the one-step policy $X_{0,1}$. We also highlight that the instantaneous velocity corresponds to the flow-matching policy can be recovered by taking the time limit yielding the tangent condition: $\lim_{r \to t} \partial_t X_{r,t}(a_r \mid s) = u_{t,t}(a_t|s) := v_t(a_t \mid s)$. As a result, this allows a supervision signal along the time diagonal $r = t$ amounting to classical flow-matching,

$$\mathcal{L}_{\text{Diag}} = \mathbb{E}_{t,q(z),p_t(a_t|z,s)} \|u_{t,t}(a_t \mid s) - (a_1 - a_0)\|_2^2. \tag{3}$$

To train the underlying flow map on the off-diagonal $r < t$, we follow the standard practice of enforcing consistency rules that are derived from satisfying the flow-map jump condition, as well as the semi-group property of the ODE [Boffi et al., 2025]. This leads to PINN style ℓ_2-regression objectives that *distill* the approximated ODE velocity field into $X_{r,t}$ by enforcing the Lagrange, Euler, and Progressive conditions of the flow map on the off-diagonal times $r < t$ combined with a stop-gradient $\texttt{sg}$:

1. **Lagrangian policy-distillation**.

$$\mathcal{L}_{\text{LPD}} = \int_0^1 \int_0^t \mathbb{E}_{p_r(a_r|z,s)} \left[\left| \partial_t X_{r,t}(a_r \mid s) - \texttt{sg}\left(u_{t,t}(X_{r,t}(a_r \mid s))\right) \right|^2 \right] dr dt, \tag{4}$$

2. **Eulerian policy-distillation**.

$$\mathcal{L}_{\text{EPD}} = \int_0^1 \int_0^t \mathbb{E}_{p_r(a_r|z,s)} \left[\left| \partial_r X_{r,t}(a_r \mid s) + \texttt{sg}\left(\nabla X_{r,t}(a_r \mid s) u_{r,r}(a_r \mid s)\right) \right|^2 \right] dr dt, \tag{5}$$

3. **Progressive policy-distillation**.

$$\mathcal{L}_{\text{PPD}} = \int_0^1 \int_0^t \mathbb{E}_{p_r(a_r|z,s)} \left[\left| X_{w,t}(X_{r,w}(a_r \mid s)) - \texttt{sg}\left(X_{r,t}(a_r \mid s)\right) \right|^2 \right] dr dt d\gamma, \tag{6}$$

   where $w = (1-\gamma)r + \gamma t$ with $\gamma \in [0,1]$.

Relation to mean-flow policies. Critically, in contrast to prior work, equating policy learning with flow-maps unlocks the entire arsenal of flow-map-based learning objectives—with mean flows policies [Geng et al., 2025, Nguyen and Yoo, 2026, Zhan et al., 2026] being a specific instantiation of the Eulerian policy. In particular, mean-flow policies can be derived as a specific instance of the Eulerian policy distillation objective outlined in eq. (5) above (see §A.2), and the instantaneous velocity constraint is an application of the tangent condition and is simply the diagonal loss in eq. (3).

3.2 Offline RL with Flow Map Policies

We now deploy flow-map policies for offline-to-online reinforcement learning within an actor-critic framework. We first pre-train an efficient flow-map actor on an existing offline dataset $\mathcal{D} = \{(s, a_1, r, s')\}$, along with a critic Q-network. We parametrize the actor as $u_{r,t}(a_r|s)$ over all time pairs $(r,t) \in [0,1]^2$, trained with the policy self-distillation objectives from §3.1: $\mathcal{L}^{\text{off}}_{\text{actor}} = \mathcal{L}_{\text{Diag}} + \lambda \mathcal{L}_{\text{SD}}$, where $\mathcal{L}_{\text{SD}}$ corresponds to any of the policy self-distillation losses and λ is a hyper-parameter that controls the strength of off-diagonal training. For maximally efficient action generation, we can simply invoke the flow-map policy $a_1 = X^{\text{off}}_{0,1}(a_0|s)$ and generate actions in a single forward pass by directly transporting the prior noisy action a_0 to the clean action a_1 in one step:

$$a_1 = a_0 + u^{\text{off}}_{0,1}(a_0 \mid s), \quad a_0 \sim \mathcal{N}(0, I). \tag{7}$$

We train the critics via clipped double Q-learning with EMA targets Q_{ϕ_j} [Fujimoto et al., 2018]:

$$\mathcal{L}_{\text{critic}}(\phi_j) = \mathbb{E}_{(s,a_1,r,s')\sim\mathcal{D}}\left[\left(Q_{\phi_j}(s,a_1) - y\right)^2\right], \quad y = r + \gamma \min_{j=1,2} Q_{\phi_j}(s', X^{\text{off}}_{0,1}(a'_0 \mid s')). \tag{8}$$

3.3 Efficient Trust-Region Based Online Adaptation

Transitioning to the online phase introduces the challenge of identifying the optimal action to imitate when training generative policies: in continuous action spaces without a curated dataset, solving $\arg\max_a Q(s,a)$ as a learning target is intractable. A common strategy is the "best-of-N" heuristic, which draws N actions from the policy and selects the one with the highest Q-value [Zhan et al., 2026]. However, this naive strategy imposes a non-trivial drawback of requiring a large number of sampled actions N, which requires at minimum N one-step simulations of the flow-map actor and Q-function evaluations. We next develop a more principled approach that finds a locally optimal improvement by constructing a trust region. Consider a flow-map policy $\pi^{\text{off}}(\cdot|s)$, the natural question for online adaptation is:

Q. *What is the optimal perturbation Δ for $a_1 \sim \pi^{off}$ that maximizes the critic's Q-function?*

To answer this question, we assume the existence of an optimal action $a_1^* = a_1 + \Delta^*$ that is feasible—i.e., reachable from the flow-map policy via a perturbation Δ. To prevent unbounded deviation from $a_1 \sim \pi^{\text{off}}(\cdot|s)$, we constrain Δ within a trust region of radius η around the critic's current Q-value. This yields the following non-linear optimization problem that maximizes the critic's Q-function:

$$\arg\max_{\Delta} \quad \mathbb{E}_{r\sim\mathcal{U}[0,1)}\left[Q_\phi\big(s, X^{\text{off}}_{r,1}(a_r \mid s) + \Delta\big)\right] \quad \text{s.t.} \qquad \|\Delta\|_2 \le \eta \tag{9}$$

In the case where the Δ-perturbation is given as the average velocity network $u_{r,1}(a_r|s)$, constraining the perturbation $\|\Delta\|_2 \le \eta$ in action space is equivalent to bounding $(1-r)\|u_{r,1}(a_r|s) - u^{\text{off}}_{r,1}(a_r|s)\|_2$. As the critic Q-function is non-linear, this optimization problem is challenging to solve in closed form. Instead, we can consider a first-order approximation of optimality that aims to find *optimal target displacement* to any generic reference $u^{\text{ref}}_{r,1}(a_r|s)$. Interestingly, under these settings, the analytic expression of the optimal target displacement admits a closed-form expression.

Theorem 3.2. *Consider a flow-map policy $\pi^{ref}(\cdot|s)$ with underlying flow map $X^{ref}_{r,1}$, generating actions $a_1 = a_r + (1-r)\, u^{ref}_{r,1}(a_r \mid s)$. The optimal average velocity $u^*_{r,1}$ that maximizes the first-order expansion of Q_ϕ around a_1, subject to trust-region constraint $\|u_{r,1} - u^{ref}_{r,1}\|_2 \le \eta$, is:*

$$u^*_{r,1}(a_r \mid s) = u^{ref}_{r,1}(a_r \mid s) + \eta \frac{\nabla_a Q_\phi(s,a_1)}{\|\nabla_a Q_\phi(s,a_1)\|_2}. \tag{10}$$

Proof Sketch. To maximize Q_ϕ over $u_{r,1}$, we substitute the flow-map parameterization into a first-order Taylor expansion around a_1, which reduces the problem to maximizing $\langle \nabla_a Q_\phi(s,a_1),\, u_{r,1} - u^{\text{ref}}_{r,1}\rangle$ subject to $\|u_{r,1} - u^{\text{ref}}_{r,1}\|_2 \le \eta$. Solving the associated KKT conditions yields the optimal closed-form solution. The full proof is provided in §A.1. □

Theorem 3.2 is exact for the linearized subproblem rather than the original non-linear problem. If $\nabla_a Q_\phi$ is L-Lipschitz, the resulting target is within $L(1-r)^2\eta^2$ of the non-linear action-space optimum and strictly improves the critic whenever $(1-r)\eta < 2\|\nabla_a Q_\phi(s,a_1)\|_2/L$ (proposition B.1). Theorem 3.2 holds for any reference flow-map policy. Setting $\pi^{\text{ref}} = \pi^{\text{off}}$, i.e., anchoring to the offline flow-map velocity $u^{\text{off}}_{r,1}(a_r|s)$, results in the following target average velocity:

$$u^*_{r,1}(a_r|s) = u^{\text{off}}_{r,1}(a_r|s) + \eta \frac{\nabla_a Q_\phi(s,a_1)}{\|\nabla_a Q_\phi(s,a_1)\|_2}. \tag{11}$$

The analytic form of eq. (11) enables us to form a learning target for efficient online adaptation that we term FLOW MAP Q-GUIDANCE (FMQ). Specifically, we construct the interpolant $a_r = (1-r)\, a_0 + r\, a_{\text{data}}$ using noise $a_0 \sim \mathcal{N}(0, I)$ and actions from a replay buffer $a_{\text{data}} \sim \mathcal{D}$. This allows to then regress $u_{r,1}(a_r|s)$ against the self-bootstrapped trust-region target below:

$$\mathcal{L}_{\text{FMQ}}(\theta) = \mathbb{E}_{r,\, a_0,\, a_{\text{data}}}\left[\left\| u^\theta_{r,1}(a_r) - \text{sg}\left(u^{\text{off}}_{r,1}(a_r) + \eta \frac{\nabla_a Q_\phi(s,a_1)}{\|\nabla_a Q_\phi(s,a_1)\|_2 + \kappa_1}\right)\right\|_2^2\right], \tag{12}$$

where $\mathrm{sg}(\cdot)$ is the stop-gradient operator, and $\kappa_1 > 0$ is a stability constant as described in algorithm 1.

Uncertainty-Aware Adaptive Trust Region. A fixed radius η applies the same step size regardless of critic's confidence. We now formulate an adaptive per-sample η-radius driven by a cheap heuristic, driven by capturing the epistemic uncertainty in the critic ensemble. Given a twin-critic ensemble, we define $\delta_{\text{critic}}(s,a) = \frac{1}{\sqrt{2}}|Q_{\phi_1}(s,a) - Q_{\phi_2}(s,a)|$ that captures the absolute discrepancy of Q-values amongst the critics. This allows us to design a batch-normalized per-sample *effective* trust region,

$$\eta_{\text{eff}}(s,a) = \frac{\eta}{1 + \beta\,\tilde{\delta}_{\text{critic}}(s,a)}, \qquad \tilde{\delta}_{\text{critic}}(s,a) = \frac{\delta_{\text{critic}}(s,a)}{\frac{1}{B}\sum_{i=1}^{B}\delta_{\text{critic}}(s_i,a_i) + \kappa_2}, \tag{13}$$

where $\kappa_2 > 0$ is a small constant, and β a hyper-parameter. By construction, $\eta_{\text{eff}} \in (0, \eta]$ decays monotonically with the magnitude $\delta_{\text{critic}}(s,a)$: a small discrepancy δ_{critic} leads to larger steps, while conversely a larger discrepancy δ_{critic} results in the prioritization of the offline flow map actor. We view η_{eff} as a practical surrogate for critic reliability rather than a formal certificate (section B).

3.4 Inference-Time Q-Guided Search

The flow map policy induces a mapping that transports noisy actions marginals $[X_{r,t}]_{\#}p_r = p_t$ for all $r, t \in [0,1]$. This mapping is fundamentally incapable of capturing the conditional posterior over endpoints that also maximize a critic's Q-value. As a result, the initial sampling of a_0 may have a disproportionate impact on the solutions to the optimization problem in eq. (9). Instead of training a separate stochastic flow map for reward alignment [Potaptchik et al., 2026, Holderrieth et al., 2026] we opt for a purely inference-time search strategy. Specifically, we next construct a stochastic sampler for flow map actors that also leverages the trust region of the critic's Q-value.

Stochastic Sampling with SNR. To design a stochastic sampler, we leverage a renoising strategy based on the signal-to-noise ratio (SNR). In particular, given the one-step flow map actor *after online adaptation* $a_1 = X_{0,1}^{\text{adapt}}(a_0|s)$, we can re-noise by judiciously selecting a new time $t' < 1$. To do so, we design the re-noising interpolant by selecting $t' = \mathrm{SNR}/(1+\mathrm{SNR}) \in (0,1)$:

$$a_{t'} = t' \cdot a_1 + (1 - t') \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \tag{14}$$

A second application of the flow map then transports this intermediate state back to time $t = 1$, i.e., $\tilde{a}_1 = X_{t',1}^{\text{adapt}}(a_{t'} \mid s)$. Crucially, each draw of ε yields a *different* action $\tilde{a}_1$—a stochastic sample from the flow map to which the trust-region update can be re-applied. The approach thus defines an iterative refinement procedure during inference: (1) we corrupt the current actions a_1 to t' using eq. (14), gaining access to diverse actions noisy intermediate states a'_t, and (2) then re-apply the optimal trust-region Q-value projection to each new sample, obtaining $\tilde{a}_1$. The noise level t' controls the exploration–exploitation balance: small t' (low SNR) places $a_{t'}$ closer to pure noise, allowing the flow map to explore distant modes. Conversely, a large t' (high SNR) preserves most of the current action a_1, it restricts the update to a more local refinement.

Q-Guided Beam Search. We now outline an inference-time search strategy that combines the stochastic sampler with beam search. This new algorithm is deployed only once at inference — i.e., inference-time scaling via search — and, as a result, does not affect training speed for online updates. We provide the full algorithmic description in algorithm 2. We instantiate this new final inference procedure Q-GUIDED BEAM SEARCH (QGBS), which balances exploration and exploitation. Specifically, QGBS operates over M particles that are refined over K steps along the already online-adapted flow map policy trajectory. In summary, the algorithm follows the following two steps iteratively:

1. *Exploration:* The first step in QGBS constitutes an exploration phase that diversifies the candidate N actions to a batch B of intermediate states that then yields a total of $\bigcup_{i=1}^{N\cdot B} X_{t',1}^{\text{adapt}}(a'_t|s)$ actions.
2. *Exploitation:* The second step selects the most promising M particles using the critic $Q_\phi(s,\tilde{a}_1)$, which are then used in the trust region update (eq. (11)), before progressing to the next beam.

After K steps, the procedure returns $\arg\max_i Q_\phi(s,a_i)$. When $K{=}0$, the method reduces to best-of-M Q-Steering (a single application of Theorem 3.2 without iteration).

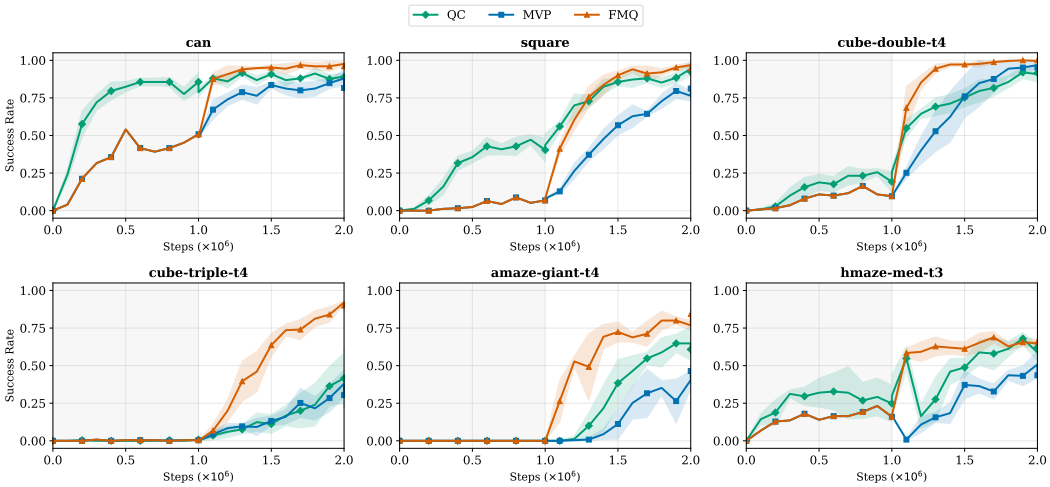


Figure 2: Training curves on 6 environments. Average success rate at every 100K during 1M offline followed by 1M online steps over 5 seeds. Shaded regions indicate 95% CIs.

4 Experiments

We investigate the application of FMQ across 12 robotic manipulation and locomotion tasks with varying difficulties across 7 environments from the OGBench [Park et al., 2025a] and Robomimic [Mandlekar et al., 2021] benchmarks. The manipulation tasks include two from Robomimic (`can`, `square`) and six from OGBench (`cube-dbl-t3/4`, `cube-trl-t3/4`, `scene-t4/5`). Locomotion is evaluated on four OGBench tasks (`hmaze-med-t3/4`, `amaze-gnt-t4/5`). During offline pre-training, we use multi-human demonstration datasets for Robomimic and the default noisy-expert datasets (play-style and navigate) for OGBench. In addition, we utilize single-task OGBench variants for offline-to-online RL. The humanoid and ant maze tasks use sparse rewards, while all others use dense rewards. For clarity, we report full training configurations and experimental setups in §J.

Baselines. We compare FMQ against two main baselines: (1) QC [Li et al., 2025] trains a multi-step flow matching policy with 10 integration steps. (2) As our second baseline, we report the state-of-the-art method MVP [Zhan et al., 2026], which trains a mean flow policy with an initial velocity constraint. For MVP, we distinguish MVP* as results taken directly from the original paper, which is only available in $6/12$ environments considered here, from our reproduction MVP, allowing us to investigate all considered environments. All baselines share the same model architecture and follow the same clipped double Q-learning algorithm [Fujimoto et al., 2018]. At inference time, QC, MVP, and our method FMQ all select actions via best-of-32 sampling, choosing the action with the highest Q-value. FMQ thus removes best-of-N only from the actor-training target; section C.3 reports evaluation with $N{=}1$ and per-action latency.

Evaluation protocol. We follow the standard offline-to-online protocol from Park et al. [2025a]: 1M gradient steps of offline pre-training using the provided dataset, followed by 1M steps of online fine-tuning with environment interaction. During the online phase, newly collected transitions are appended to the replay buffer. To monitor training, we evaluate the policy every 100K steps over 50 episodes with randomized initial states. Finally, we evaluate the last checkpoint across 50 unseen test episodes per environment and compute the average success rate across 5 seeds and report the Interquartile Mean (IQM) alongside 95% stratified bootstrap confidence intervals [Agarwal et al., 2021].

4.1 Main Results

We report our main quantitative results in table 1 and observe that FMQ achieves the highest IQM score $(0.91; [0.89, 0.93])$, outperforms MVP by 21.3% $(0.75; [0.73, 0.77])$ and QC by 5.8% $(0.86; [0.84, 0.87])$. The improvement is most pronounced on challenging environments: on `cube-trl-t4`, FMQ reaches 0.88 compared to 0.37 for QC and 0.32 for MVP, and on `amaze-gnt-t4`, FMQ achieves 0.80 compared to 0.64 and 0.42, respectively. We also note that QC outperforms MVP on average (0.76 vs. 0.68), but this comes at $10\times$ computational overhead at inference due to simulation of the flow rather than one-step generation. Nevertheless, we find

Table 1: Success rate (mean ± std over 5 seeds, 50 episodes). Best per row in **bold**, second best underlined. Aggregate performance is measured by the IQM scores with 95% CIs.

Environment	QC	MVP*	MVP	MVP + QGBS (Ours)	FMQ (Ours)	FMQ + QGBS (Ours)
`can`	0.88 ± 0.06	0.92 ± 0.07	0.83 ± 0.07	0.87 ± 0.07	0.96 ± 0.04	$\mathbf{0.97 \pm 0.03}$
`square`	0.89 ± 0.04	0.93 ± 0.01	0.82 ± 0.04	0.83 ± 0.05	0.94 ± 0.02	$\mathbf{0.95 \pm 0.04}$
`cube-dbl-t3`	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
`cube-dbl-t4`	0.92 ± 0.05	0.95 ± 0.04	0.98 ± 0.02	0.98 ± 0.02	0.98 ± 0.02	$\mathbf{1.00 \pm 0.00}$
`cube-trl-t3`	0.83 ± 0.08	0.71 ± 0.06	0.64 ± 0.12	0.78 ± 0.12	0.78 ± 0.10	$\mathbf{0.84 \pm 0.04}$
`cube-trl-t4`	0.37 ± 0.26	0.52 ± 0.11	0.32 ± 0.07	0.37 ± 0.09	$\mathbf{0.88 \pm 0.07}$	0.87 ± 0.05
`scene-t4`	0.99 ± 0.01	—	0.92 ± 0.02	0.98 ± 0.02	$\mathbf{1.00 \pm 0.00}$	0.99 ± 0.01
`scene-t5`	0.96 ± 0.02	—	0.90 ± 0.06	0.95 ± 0.05	0.98 ± 0.02	$\mathbf{1.00 \pm 0.00}$
`hmaze-med-t3`	0.65 ± 0.11	—	0.47 ± 0.10	0.53 ± 0.03	$\mathbf{0.69 \pm 0.04}$	0.58 ± 0.07
`hmaze-med-t4`	0.04 ± 0.03	—	0.00 ± 0.00	0.02 ± 0.02	$\mathbf{0.06 \pm 0.03}$	$\mathbf{0.06 \pm 0.03}$
`amaze-gnt-t4`	0.64 ± 0.12	—	0.42 ± 0.06	0.43 ± 0.04	$\mathbf{0.80 \pm 0.06}$	0.77 ± 0.03
`amaze-gnt-t5`	0.91 ± 0.05	—	0.82 ± 0.08	0.90 ± 0.06	$\mathbf{0.92 \pm 0.04}$	$\mathbf{0.92 \pm 0.05}$
IQM SR [95% CI]	0.86 [0.84, 0.87]	—	0.75 [0.73, 0.77]	0.81 [0.78, 0.83]	0.91 [0.89, 0.93]	**0.93** [**0.91**, **0.94**]

that FMQ outperforms QC in $10/12$ environments using only a single generation step. These results demonstrate the benefit of the closed-form trust-region Q-guidance in FMQ in comparison to best-of-N. Against the results originally reported by MVP (MVP*), FMQ matches or exceeds MVP* on all six shared tasks (mean 0.92 vs. 0.84; 10.1% relative), whereas the 21.3% margin refers to our reproduction (section C.1).

Inference scaling. To evaluate our complementary contribution Q-GUIDED BEAM SEARCH (QGBS) that can be used as a stochastic sampler on any flow-map policy, including the baseline method MVP. We introduce two additional configurations, MVP + QGBS and FMQ + QGBS. Specifically, we replace the best-of-N sampling in inference time with our stochastic sampling algorithm, which is combined with beam search ($K = 1$, $B = 4$, $M = 4$) and outlined in §3.4. Overall, applying QGBS we observe a relative increase in IQM by 8.0% (from 0.75 to 0.81) for MVP and a relative increase by 2.2% (from 0.91 to 0.93) for FMQ. However, for a small number of locomotion tasks (`hmaze-med-t3` and `amaze-gnt-t4`), QGBS degrades performance by 15.9% and 3.8% respectively. These are among the tasks most sensitive to critic-gradient noise (section C.2): as QGBS ranks candidates with the critic, it can be disabled when critic-ensemble disagreement is high (section C.4). Overall, we find that combining both our proposed training and inference algorithms FMQ + QGBS leads to the best performance, achieving the highest IQM of 0.93; $[0.91, 0.94]$, with non-overlapping confidence intervals against all baselines, including QC (0.86; $[0.84, 0.87]$) and MVP (0.75; $[0.73, 0.77]$). These results highlight the impact of performance by increasing the compute budget at inference through stochastic sampling and beam search.

4.2 Sample Efficiency Analysis

We next investigate the sample efficiency gains of using Q-guidance to train our flow map policies. In fig. 2, we plot the training curves during the online adaptation for all methods across 6 environments (see fig. 17 for full). We find that FMQ consistently converges faster than MVP during online fine-tuning, despite both methods starting from the same offline checkpoint. To quantify this advantage, we define ξ as the highest success rate that MVP and FMQ can reach, computed per seed and environment. In table 12, we measure speedup T: the number of steps to first reach $\{75\%, 85\%, 95\%, 100\%\}$ of ξ. The speedup ratio $T_{\text{MVP}}/T_{\text{FMQ}}$, averaged over 5 seeds, quantifies how many times faster FMQ converges to each fraction of ξ. In the online phase (1M–2M), FMQ reaches the highest success rate achievable by MVP $2.77\times$ faster on average, and up to $6.14\times$ on `hmaze-med-t3`. These results further confirm that Q-gradient alignment provides a stronger learning signal than best-of-N selection, leading to faster policy improvement per environment step.

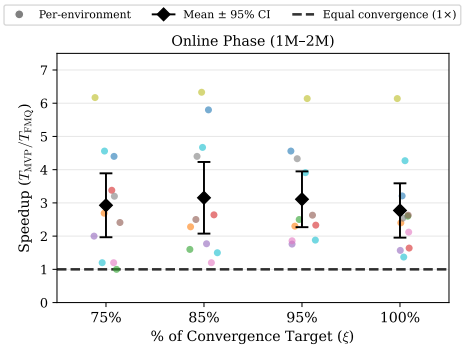


Figure 3: Convergence speedup of FMQ compared to MVP at success targets (ξ), with 95% CIs.

4.3 Ablation Studies

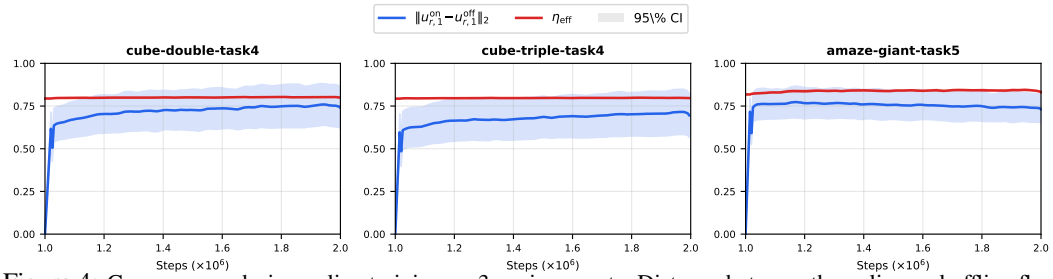


Figure 4: Convergence during online training on 3 environments. Distance between the online and offline flow map policies (blue) converges to the adaptive radius η_{eff} (red) as the policy incorporates the Q-guidance.

Inference-time Beam Search. The computational cost of utilizing QGBS is NFE $= M(1 + KB)$ per action selection, where M is the number of initial candidates, K is the number of re-noising steps, and B is the number of completions per candidate. We note that best-of-N sampling corresponds to K=0 and M=N. In table 2, we show that the optimal configuration (K=1, B=4, M=4) achieves a peak IQM of 0.93 with only 20 FE—37.5% fewer than best-of-32—suggesting that diversifying candidates through renoising is more efficient than considering more candidates. Increasing K beyond 1 does not improve performance, suggesting that only a modest increase in inference cost is needed for optimal performance.

Table 2: QGBS efficiency ablation.

K	$\{B, M\}$	NFE	IQM
0	$\{1, 32\}$	32	0.91 $[0.89, 0.93]$
1	$\{4, 4\}$	20	**0.93** $[\mathbf{0.91}, \mathbf{0.94}]$
1	$\{2, 8\}$	24	**0.93** $[\mathbf{0.91}, \mathbf{0.95}]$
1	$\{1, 16\}$	32	0.92 $[0.90, 0.93]$
1	$\{4, 16\}$	80	0.92 $[0.90, 0.93]$
2	$\{4, 4\}$	36	0.91 $[0.90, 0.93]$
2	$\{1, 16\}$	48	0.90 $[0.89, 0.92]$
2	$\{4, 16\}$	144	0.90 $[0.89, 0.91]$

Trust-Region Convergence Analysis. We evaluate the impact of trust-region in eq. (12). Specifically, we measure the distance $\|u^{\text{on}}_{r,1} - u^{\text{off}}_{r,1}\|_2$ relative to the frozen offline policy. At the onset of online training, the online and offline velocity fields coincide, and the distance is 0. As online training begins, the learning actively drives $u^{\text{on}}_{r,1}$ toward $u^{\text{off}}_{r,1} + \eta \nabla_a Q_\phi(s, a_1) / \|\nabla_a Q_\phi(s, a_1)\|_2$—as evidenced in fig. 4—and stabilizes near η_{eff}. Thus, the flow map policy incorporates the normalized Q-gradient direction while remaining safely constrained within the trust-region radius (c.f. fig. 18 for all environments).

Dependence on critic. Critic accuracy is a central challenge in offline-to-online RL. FMQ addresses it by design as the trust region caps each update at η_{eff}, and the adaptive radius shrinks where the ensemble disagrees. To test this, we design two ablations (section C.2). First, at the beginning of online adaptation the critic is at its worst and Q-values are the most biased; yet, mean success increases from 18% at the offline checkpoint to 61% at the first 100k online checkpoint, versus 38% for MVP and 58% for QC. Thus, we observe no sustained transition drop. Second, we inject c std of Gaussian noise into the critic gradient at every online policy update. Even at $c = 1.0$, where noise equals the true gradient in magnitude, no task collapses and average success declines from 0.62 to 0.54 on the six hardest tasks.

Efficiency analysis with wall clock time. We position FMQ as a training-time method, which leads to significantly faster and better training, while the evaluation stage is shared with other one-step baselines like MVP. The total online run time of FMQ (149 min) matches that of MVP (151 min) and is 1.54× lower than that of QC (229 min), confirming the closed-form Q-gradient target adds no per-step wall clock overhead, and FMQ reaches every performance milestone 2.1–3.0× sooner in wall-clock time than both baselines. At inference, FMQ and MVP have identical cost (0.23 vs. 0.22 ms at N=1), so the accuracy gains of FMQ come at zero additional latency (section C.3). Ablations of the trust-region radius, the disagreement coefficient β, and gradient normalization are reported in section C.5.

Flow map policy variants. We next ablate the offline flow map policy variants and their impact on performance. In table 3. ESD and LSD achieve the same IQM of 0.79, while PSD lags slightly at 0.77. However, ESD exhibits the narrowest 95% bootstrap CI $[0.77, 0.81]$, indicating it is the most consistent across environments. We therefore adopt ESD as the default formulation for our presented results.

5 Related work

Generative Policies. Diffusion models and flow matching have emerged as expressive policy representations [Chi et al., 2023, Pearce et al., 2023]. For policy learning, prior methods train diffusion and flow-matching models via weighted behavioral cloning [Lu et al., 2023, Kang et al., 2023], reparameterized policy gradients [Wang et al., 2023, Ding and Jin, 2024, Zhang

et al., 2024], rejection sampling [Chen et al., 2024, Hansen-Estruch et al., 2023, He et al., 2024], and latent-space optimization with a generative decoder [Zhang et al., 2026]. While effective, reparameterized gradients require costly backpropagation through time (BPTT). To address the latency of flow-matching multi-step models, FQL [Park et al., 2025b] distills a multi-step flow into a separate one-step student network, while QC [Li et al., 2025] groups action sequences. MVP [Zhan et al., 2026] natively achieves one-step generation but uses a "generate-and-select" heuristic to find imitation targets. Concurrently, RFM [Li et al., 2026] seeks to target the unnormalized Q-distribution with posterior mean estimates during training. In contrast, we cast one-step policies as flow maps and adapt them with FMQ, a closed-form Q-guided target that solves a linearized trust-region problem.

Offline-to-Online RL. Offline-to-online RL accelerates online learning by initializing from a static dataset [Levine et al., 2020]. However, offline RL must contend with overestimation of Q-values for out-of-distribution actions, addressed through divergence penalties [Fujimoto et al., 2019, Wu et al., 2019, Nair et al., 2020, Wang et al., 2023], pessimistic value estimates [Kumar et al., 2020, Yu et al., 2020, An et al., 2021], or in-sample maximization [Kostrikov et al., 2022, Garg et al., 2023]. When transitioning to the online phase, distribution shift can cause catastrophic forgetting of the behavioral prior [Lee et al., 2022, Song et al., 2023, Nakamoto et al., 2023]. Recent state-of-the-art flow-matching methods rely on behavioral regularization [Park et al., 2025b] or best-of-N selection [Li et al., 2025, Zhan et al., 2026] to stabilize adaptation. In contrast, we formulate online fine-tuning as a trust-region problem with FMQ.

Table 3: Self-distillation loss ablation.

Environment	ESD	PSD	LSD
cube-trl-t3	0.79 ± 0.06	0.74 ± 0.13	$\mathbf{0.84 \pm 0.09}$
cube-trl-t4	$\mathbf{0.90 \pm 0.05}$	0.87 ± 0.09	0.88 ± 0.06
hmaze-med-t3	$\mathbf{0.64 \pm 0.02}$	0.48 ± 0.29	0.62 ± 0.09
hmaze-med-t4	0.03 ± 0.03	0.04 ± 0.04	$\mathbf{0.06 \pm 0.00}$
amaze-gnt-t4	0.82 ± 0.02	$\mathbf{0.86 \pm 0.04}$	0.78 ± 0.06
amaze-gnt-t5	$\mathbf{0.92 \pm 0.01}$	0.87 ± 0.05	0.91 ± 0.03
IQM	**0.79**	0.77	**0.79**
95% CI	[0.77, 0.81]	[0.70, 0.81]	[0.75, 0.82]

Q-gradient and proximal actor updates. Actor-gradient methods [Silver et al., 2014, Fujimoto et al., 2018] and Diffusion-QL [Wang et al., 2023] backpropagate the critic through the actor or its sampling chain, whereas DIPO [Yang et al., 2023] iteratively shifts replay-buffer actions along the critic gradient and Q-score matching [Psenka et al., 2024] aligns the policy score with the action-gradient of the critic. Proximal methods constrain policy updates via KL trust regions [Schulman et al., 2015], behavior regularization [Fujimoto and Gu, 2021, Nair et al., 2020], or evolving regularizers [Li et al., 2023]. Concurrently, FlowDPG [Shi et al., 2026] distills a normalized critic-gradient correction into a velocity field. In contrast, FMQ regresses a flow map onto a single, reference-anchored and uncertainty-adaptive Cauchy step [Nocedal and Wright, 2006, Xie et al., 2024] from the frozen offline policy, without backpropagating the critic through the actor.

6 Conclusion

In this paper, we bridge the gap between expressive generative policies and the low-latency requirements of offline-to-online RL. We formulate online adaptation of one-step flow map policies as a trust-region optimization problem, yielding FLOW MAP Q-GUIDANCE (FMQ): a closed-form, locally optimal Q-guided update that improves a linearized critic while remaining anchored to the offline behavioral prior. We further introduce QGBS, an inference-time refinement procedure based on stochastic renoising and beam search that improves any flow map policy. Across 12 continuous-control tasks from OGBench and RoboMimic, FMQ establishes state-of-the-art performance, outperforming our reproduction of the previous leading one-step policy, MVP, by a 21.3% relative margin in IQM success.

While FMQ enjoys an efficient linearized critic approximation, its effectiveness depends on critic accuracy and local linearity: our improvement guarantees are conditional on critic smoothness and gradient alignment, critic disagreement is only a practical surrogate for these conditions, and QGBS can degrade performance when the critic ranks candidates unreliably. Extending flow map adaptation with curvature-aware updates, stronger uncertainty estimates, and deployment on physical robotic platforms are promising directions for future work.

Acknowledgments and Disclosure of Funding

We thank Jorge de Freitas for their helpful comments on an earlier draft and Luca Eyring for the insightful discussion on renoising. This work was supported by UKRI (EP/Y037111/1) as part of the ProSafe project (EU Horizon 2020, MSCA, grant no. 101119358). The authors declare no competing interests.

References

R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021. (Cited on page 7)

M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. (Cited on page 18)

M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023. URL https://arxiv.org/abs/2303.08797. (Cited on pages 1 and 3)

G. An, S. Moon, J.-H. Kim, and H. O. Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021. (Cited on page 10)

N. M. Boffi, M. S. Albergo, and E. Vanden-Eijnden. Flow map matching. *arXiv preprint arXiv:2406.07507*, 2, 2024. (Cited on pages 2 and 18)

N. M. Boffi, M. S. Albergo, and E. Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation, 2025. URL https://arxiv.org/abs/2505.18825. (Cited on pages 2 and 4)

T. Chen, Z. Wang, and M. Zhou. Diffusion policies creating a trust region for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:50098–50125, 2024. (Cited on page 10)

C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. (Cited on page 9)

C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. (Cited on page 1)

O.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 1(2), 2023. (Cited on page 1)

Z. Ding and C. Jin. Consistency models as a rich and efficient policy class for reinforcement learning. In *International Conference on Learning Representations*, 2024. (Cited on page 9)

J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. (Cited on page 1)

S. Fujimoto and S. S. Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 20132–20145, 2021. (Cited on pages 1 and 10)

S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:3544558. (Cited on pages 5, 7, 10, and 28)

S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019. (Cited on page 10)

D. Garg, J. Hejna, M. Geist, and S. Ermon. Extreme q-learning: Maxent rl without entropy. In *International Conference on Learning Representations*, 2023. (Cited on page 10)

Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He. Mean flows for one-step generative modeling, 2025. URL https://arxiv.org/abs/2505.13447. (Cited on page 4)

P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023. (Cited on page 10)

L. He, L. Shen, and X. Wang. Aligniql: Policy alignment in implicit q-learning through constrained optimization. *arXiv preprint arXiv:2405.18187*, 2024. (Cited on page 10)

P. Holderrieth, D. Chen, L. Eyring, I. Shah, G. Anantharaman, Y. He, Z. Akata, T. Jaakkola, N. M. Boffi, and M. Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. *arXiv preprint arXiv:2602.05993*, 2026. (Cited on page 6)

N. Jaques, A. Ghandeharioun, J. H. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019. (Cited on page 1)

G. Kahn, A. Villaflor, P. Abbeel, and S. Levine. Composable action-conditioned predictors: Flexible off-policy learning for robot navigation. In *Conference on robot learning*, pages 806–816. PMLR, 2018. (Cited on page 1)

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002. (Cited on page 18)

B. Kang, X. Ma, C. Du, T. Pang, and S. Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:67195–67212, 2023. (Cited on page 9)

I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. (Cited on page 10)

A. Kumar. Data-driven deep reinforcement learning. *Berkeley Artificial Intelligence Research (BAIR), Tech. Rep*, 2019. (Cited on page 1)

A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191, 2020. (Cited on page 10)

S. Lee, Y. Seo, K. Lee, P. Abbeel, and J. Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1702–1712. PMLR, 2022. (Cited on page 10)

S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. (Cited on pages 1 and 10)

J. Li, X. Hu, H. Xu, J. Liu, X. Zhan, and Y.-Q. Zhang. PROTO: Iterative policy regularized offline-to-online reinforcement learning. *arXiv preprint arXiv:2305.15669*, 2023. (Cited on page 10)

Q. Li, Z. Zhou, and S. Levine. Reinforcement learning with action chunking. *arXiv preprint arXiv:2507.07969*, 2025. (Cited on pages 7, 10, 28, and 35)

Z. Li, S. Tang, and N. Azizan. Reverse flow matching: A unified framework for online reinforcement learning with diffusion and flow policies. In *International Conference on Machine Learning*, 2026. (Cited on page 10)

Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. (Cited on pages 1 and 3)

X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. (Cited on pages 1 and 3)

C. Lu, H. Chen, J. Chen, H. Su, C. Li, and J. Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2023. (Cited on page 9)

A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, 2021. (Cited on page 7)

A. Nair, A. Gupta, M. Dalal, and S. Levine. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020. (Cited on page 10)

M. Nakamoto, Y. Zhai, A. Singh, M. S. Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. (Cited on page 10)

T. X. Nguyen and C. D. Yoo. One-step flow q-learning: Addressing the diffusion policy bottleneck in offline reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. (Cited on page 4)

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, 2 edition, 2006. (Cited on pages 10 and 17)

S. Park, K. Frans, B. Eysenbach, and S. Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025a. (Cited on page 7)

S. Park, Q. Li, and S. Levine. Flow q-learning. In *International Conference on Machine Learning (ICML)*, 2025b. (Cited on pages 2, 10, and 21)

T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann, and S. Devlin. Imitating human behaviour with diffusion models. In *International Conference on Learning Representations*, 2023. (Cited on page 9)

S. Peluchetti. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023. (Cited on page 3)

P. Potaptchik, A. Saravanan, A. Mammadov, A. Prat, M. S. Albergo, and Y. W. Teh. Meta flow maps enable scalable reward alignment. *arXiv preprint arXiv:2601.14430*, 2026. (Cited on page 6)

M. Psenka, A. Escontrela, P. Abbeel, and Y. Ma. Learning a diffusion model policy from rewards via Q-score matching. In *International Conference on Machine Learning*, 2024. (Cited on page 10)

J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. In *International Conference on Machine Learning*, 2015. (Cited on pages 10 and 17)

K. Shi, J. Shi, P. Hebbar, Z. Zhao, T. Amarnath, Y. Su, S. Bahl, and D. Pathak. FlowDPG: Deterministic policy gradient on flow matching policies for real-world manipulation. In *Conference on Robot Learning*, 2026. (Cited on page 10)

D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, 2014. (Cited on page 10)

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015. (Cited on page 1)

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. (Cited on page 1)

Y. Song, Y. Zhou, A. Sekhari, J. A. Bagnell, A. Krishnamurthy, and W. Sun. Hybrid rl: Using both offline and online data can make rl efficient. In *International Conference on Learning Representations*, 2023. (Cited on page 10)

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. (Cited on page 3)

D. Tarasov, V. Kurenkov, A. Nikulin, and S. Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023. (Cited on page 1)

A. Terpin, N. Lanzetti, B. Yardim, F. Dörfler, and G. Ramponi. Trust region policy optimization with optimal transport discrepancies: Duality and algorithm for continuous actions. In *Advances in Neural Information Processing Systems*, volume 35, 2022. (Cited on page 18)

A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023. (Cited on page 3)

Z. Wang, J. J. Hunt, and M. Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations*, 2023. (Cited on pages 1, 9, and 10)

Y. Wu, G. Tucker, and O. Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019. (Cited on page 10)

C. Xie, C. Li, C. Zhang, Q. Deng, D. Ge, and Y. Ye. Trust region methods for nonconvex stochastic optimization beyond Lipschitz smoothness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024. (Cited on pages 10 and 17)

L. Yang, Z. Huang, F. Lei, Y. Zhong, Y. Yang, C. Fang, S. Wen, B. Zhou, and Z. Lin. Policy representation via diffusion probability model for reinforcement learning. *arXiv preprint arXiv:2305.13122*, 2023. (Cited on pages 2 and 10)

T. Yu, G. Thomas, L. Yu, S. Ermon, J. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142, 2020. (Cited on page 10)

G. Zhan, Y. Jiang, S. E. Li, Y. Lyu, X. Zhang, and Y. Yin. A transformation-aggregation framework for state representation of autonomous driving systems. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):7311–7322, 2024. (Cited on page 2)

G. Zhan, X. An, Y. Jiang, J. Duan, H. Zhao, and S. E. Li. Physics informed neural pose estimation for real-time shape reconstruction of soft continuum robots. *IEEE Robotics and Automation Letters*, 2025. (Cited on page 2)

G. Zhan, L. Tao, P. Wang, Y. Wang, Y. Li, Y. Chen, H. Li, M. Tomizuka, and S. E. Li. Mean flow policy with instantaneous velocity constraint for one-step action generation. *arXiv preprint arXiv:2602.13810*, 2026. (Cited on pages 2, 4, 5, 7, 10, 16, 28, and 35)

C. Zhang, Z. Wan, F. Chen, F. Yang, L. Feng, Y. Zhou, X. Yu, Y. You, I. Tsang, and B. An. Generative online reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 158816–158837. PMLR, 2026. (Cited on page 10)

R. Zhang, Z. Luo, J. Sjölund, T. B. Schön, and P. Mattsson. Entropy-regularized diffusion policy with q-ensembles for offline reinforcement learning. *Advances in neural information processing systems*, 37:98871–98897, 2024. (Cited on page 9)

Z. Zhu, H. Zhao, H. He, Y. Zhong, S. Zhang, H. Guo, T. Chen, and W. Zhang. Diffusion models for reinforcement learning: A survey. *arXiv preprint arXiv:2311.01223*, 2023. (Cited on page 1)

A Theoretical details

A.1 Proofs

Theorem 3.2. *Consider a flow-map policy $\pi^{ref}(\cdot|s)$ with underlying flow map $X^{ref}_{r,1}$, generating actions $a_1 = a_r + (1-r)\, u^{ref}_{r,1}(a_r \mid s)$. The optimal average velocity $u^*_{r,1}$ that maximizes the first-order expansion of Q_ϕ around a_1, subject to trust-region constraint $\|u_{r,1} - u^{ref}_{r,1}\|_2 \le \eta$, is:*

$$u^*_{r,1}(a_r \mid s) = u^{ref}_{r,1}(a_r \mid s) + \eta \, \frac{\nabla_a Q_\phi(s, a_1)}{\|\nabla_a Q_\phi(s, a_1)\|_2}. \tag{10}$$

Proof. We seek the optimal average velocity $u_{r,1}(a_r \mid s)$ that generates action

$$\bar{a}_1 = a_r + (1-r)\, u_{r,1}(a_r \mid s). \tag{15}$$

Let $u_{r,1}(a_r \mid s)$ denote a candidate average velocity, generating action $\bar{a}_1 = a_r + (1-r)u_{r,1}(a_r \mid s)$. We take the first-order Taylor expansion of the Q-function around the reference action a_1:

$$Q_\phi(s, \bar{a}_1) \approx Q_\phi(s, a_1) + \langle \nabla_a Q_\phi(s,\, a_1),\, \bar{a}_1 - a_1 \rangle \tag{16}$$

Substituting the flow map parameterization, the starting state a_r cancels out, leading to the difference in average velocity vectors:

$$\bar{a}_1 - a_1 = \big(a_r + (1-r)u_{r,1}(a_r \mid s)\big) - \big(a_r + (1-r)u^{\text{ref}}_{r,1}(a_r \mid s)\big) = (1-r)(u_{r,1}(a_r \mid s) - u^{\text{ref}}_{r,1}(a_r \mid s)) \tag{17}$$

Since $Q_\phi(s, a_1)$ is constant with respect to $u_{r,1}(a_r \mid s)$, maximizing the linear approximation subject to the trust-region constraint on the average velocity is formulated as:

$$\begin{aligned} \min_{u_{r,1}} \quad & f_0(u_{r,1}(a_r \mid s)) = -\langle \nabla_a Q_\phi(s,\, a_1),\, u_{r,1}(a_r \mid s) - u^{\text{ref}}_{r,1}(a_r \mid s) \rangle \\ \text{subject to} \quad & f_1(u_{r,1}(a_r \mid s)) = \tfrac{1}{2}\|u_{r,1}(a_r \mid s) - u^{\text{ref}}_{r,1}(a_r \mid s)\|_2^2 - \tfrac{1}{2}\eta^2 \le 0 \end{aligned} \tag{18}$$

Because $\eta > 0$, the interior of the feasible set is non-empty, satisfying Slater's constraint qualification. Therefore, strong duality holds and the KKT conditions are necessary and sufficient. The Lagrangian is:

$$\begin{aligned} \mathcal{L}(u_{r,1}, \lambda) = & -\langle \nabla_a Q_\phi(s,\, a_1),\, u_{r,1}(a_r \mid s) - u^{\text{ref}}_{r,1}(a_r \mid s) \rangle \\ & + \lambda \left(\tfrac{1}{2}\|u_{r,1}(a_r \mid s) - u^{\text{ref}}_{r,1}(a_r \mid s)\|_2^2 - \tfrac{1}{2}\eta^2 \right) \end{aligned} \tag{19}$$

Let $u^*_{r,1}(a_r \mid s)$ and λ^* be the primal and dual optima. The KKT conditions are:

$$f_1(u^*_{r,1}(a_r \mid s)) \le 0 \qquad \text{(Primal feasibility)} \tag{20}$$

$$\lambda^* \ge 0 \qquad \text{(Dual feasibility)} \tag{21}$$

$$\lambda^* f_1(u^*_{r,1}(a_r \mid s)) = 0 \qquad \text{(Complementary slackness)} \tag{22}$$

$$-\nabla_a Q_\phi(s,\, a_1) + \lambda^*(u^*_{r,1}(a_r \mid s) - u^{\text{ref}}_{r,1}(a_r \mid s)) = 0 \qquad \text{(Stationarity)} \tag{23}$$

Assuming $\nabla_a Q_\phi(s,\, a_1) \neq 0$, stationarity equation 23 requires $\lambda^* > 0$. Complementary slackness equation 22 then forces the constraint to be active:

$$\|u^*_{r,1}(a_r \mid s) - u^{\text{ref}}_{r,1}(a_r \mid s)\|_2 = \eta \tag{24}$$

Taking the norm of the stationarity condition gives $\lambda^* \eta = \|\nabla_a Q_\phi(s,\, a_1)\|_2$, so $\lambda^* = \|\nabla_a Q_\phi(s,\, a_1)\|_2 / \eta$. Substituting λ^* back into the stationarity condition yields:

$$u^*_{r,1}(a_r \mid s) = u^{\text{ref}}_{r,1}(a_r \mid s) + \eta \frac{\nabla_a Q_\phi(s, a_1)}{\|\nabla_a Q_\phi(s, a_1)\|_2} \tag{25}$$

□

A.2 Equivalence of Eulerian and Mean Flow Policies

In this section, we elucidate the equivalence between Mean Flow Policies [Zhan et al., 2026] and the Eulerian Policy in eq. (5). We begin by stating the Mean Flow Policy and its loss gradient with respect to parameters θ.

$$\mathcal{L}_{\text{MF}} = \mathbb{E}\left[\left| u^\theta_{r,t}(a_r \mid s) - \texttt{sg}\left(u^\theta_{r,t}(a_r \mid s) + (t-r)\frac{\mathrm{d}}{\mathrm{d}r}u^\theta_{r,t}(a_r \mid s)\right)\right|^2\right] \tag{26}$$

$$\nabla_\theta \mathcal{L}_{\text{MF}} = 2\mathbb{E}\left[\nabla_\theta u^\theta_{r,t}(a_r \mid s)^T \left(u^\theta_{r,t}(a_r \mid s) - \texttt{sg}\left(u^\theta_{r,t}(a_r \mid s) + (t-r)\frac{\mathrm{d}}{\mathrm{d}r}u^\theta_{r,t}(a_r \mid s)\right)\right)\right],$$

where the expectation is taken with respect to $(r, t, p_r(a_r \mid z, s))$. Now let us recall the Eulerian objective with a flow map policy parametrization $X_{r,t}(a_r \mid s) = a_r + (t-r)u^\theta_{r,t}(a_r \mid s)$ with explicit parameters θ for the average velocity:

$$\mathcal{L}_{\text{EPD}}(\theta) = \mathbb{E}\left[\left|\partial_r X_{r,t}(a_r \mid s) + \texttt{sg}\left(\nabla X_{r,t}(a_r \mid s) u^\theta_{r,r}(a_r \mid s)\right)\right|^2\right], \tag{27}$$

Let us examine the terms inside the squared norm and remove the stop-gradient operator sg. We compute the partial derivative with respect to the start time r:

$$\partial_r X_{r,t}(a_r \mid s) = -u^\theta_{r,t}(a_r \mid s) + (t-r)\partial_r u^\theta_{r,t}(a_r \mid s). \tag{28}$$

Plugging this back into the Eulerian objective, we have,

$$\mathcal{L} = \mathbb{E}\left[\left|\underbrace{-u^\theta_{r,t}(a_r \mid s)}_{T_1} + \underbrace{(t-r)\partial_r u^\theta_{r,t}(a_r \mid s) + \nabla X_{r,t}(a_r \mid s) u^\theta_{r,r}(a_r \mid s)}_{T_2}\right|^2\right]. \tag{29}$$

Applying a stop-gradient to T_2 and taking parameter gradients, Plugging this back into the Eulerian objective, we have,

$$\nabla_\theta \mathcal{L}(\theta) = 2\mathbb{E}\left[u^\theta_{r,t}(a_r \mid s) - \nabla_\theta u^\theta_{r,t}(a_r \mid s) \cdot \left(u^\theta_{r,t}(a_r \mid s) + \texttt{sg}\left(T_2\right)\right)\right]. \tag{30}$$

Now expanding the spatial gradient term in T_2, that is $\nabla X_{r,t}(a_r \mid s) u^\theta_{r,r}(a_r \mid s)$:

$$\nabla X_{r,t}(a_r \mid s) u^\theta_{r,r}(a_r \mid s) = u^\theta_{r,r}(a_r \mid s) + (t-r)\nabla u^\theta_{r,t}(a_r \mid s) u^\theta_{r,r}(a_r \mid s). \tag{31}$$

Now by invoking the tangent condition and replacing $u^\theta_{r,r}$ with the ground truth instantaneous velocity v^* we can expand T_2 have

$$T_2 = (t-r)\partial_r u^\theta_{r,t}(a_r \mid s) + v^*_r(a_r \mid s) + (t-r)\nabla u^\theta_{r,t}(a_r \mid s) v^*_r(a_r \mid s).$$

Rearranging terms and grouping $(t-r)$ terms, we notice the total derivative $\mathrm{d}/\mathrm{d}r$ corresponds exactly to T_2. We now leverage and rewrite eq. (30) succinctly:

$$\nabla_\theta \mathcal{L}(\theta) = 2\mathbb{E}\left[\nabla_\theta u^\theta_{r,t}(a_r \mid s)^T \left(u^\theta_{r,t}(a_r \mid s) - \texttt{sg}\left(u^\theta_{r,t}(a_r \mid s) + (t-r)\frac{\mathrm{d}}{\mathrm{d}r}u_{r,t}(a_r \mid s)\right)\right)\right].$$

This loss gradient matches the Mean Flow Policies' loss gradient, with the main distinction being the usage of the ground truth velocity v^*_r as opposed to the network's prediction $u^\theta_{r,r}$. Furthermore, the instantaneous velocity constraint is equivalent to the diagonal loss of eq. (3). This demonstrates that Mean Flow policies [Zhan et al., 2026] are not an independent paradigm, but rather a specific instantiation of the broader Eulerian Policy Distillation framework.

B Additional Theoretical Results

Theorem 3.2 exactly solves the first-order trust-region subproblem, rather than the original non-linear critic-maximization problem. In this section, we clarify the relationship between the velocity- and action-space constraints, relate the FMQ target to the non-linear action-space problem, give a conditional reference-policy improvement guarantee, and clarify the sense in which FMQ is a trust-region method.

Velocity- and action-space constraints. For fixed (s, r, a_r), let $a^{\mathrm{ref}} = a_r + (1-r)\, u^{\mathrm{ref}}_{r,1}(a_r \mid s)$. A velocity perturbation δu produces the endpoint displacement

$$h = \left[a_r + (1-r)\left(u^{\mathrm{ref}}_{r,1} + \delta u\right)\right] - a^{\mathrm{ref}} = (1-r)\,\delta u.$$

Therefore,

$$\|\delta u\|_2 \le \eta \quad \Longleftrightarrow \quad \|h\|_2 \le \rho_r, \qquad \rho_r := (1-r)\,\eta,$$

rather than an action-space trust region with the same radius η. Equivalently, imposing a fixed action-space radius ρ would require the r-dependent velocity radius $\eta_r = \rho/(1-r)$.

Proposition B.1 (Non-linear guarantee for the FMQ Cauchy step)**.** *Fix s, $r \in [0, 1)$, and a_r, and define*

$$a^{\mathrm{ref}} = a_r + (1-r)\, u^{\mathrm{ref}}_{r,1}(a_r \mid s), \qquad q(a) := Q_\phi(s, a), \qquad g := \nabla q(a^{\mathrm{ref}}).$$

Suppose $g \neq 0$ and ∇q is L-Lipschitz on $B(a^{\mathrm{ref}}, \rho_r)$, where $\rho_r = (1-r)\,\eta$. Define

$$\delta u_{\mathrm{FMQ}} = \eta\, \frac{g}{\|g\|_2}, \qquad h_{\mathrm{FMQ}} = (1-r)\,\delta u_{\mathrm{FMQ}} = \rho_r\, \frac{g}{\|g\|_2},$$

and let $h^\star \in \arg\max_{\|h\|_2 \le \rho_r} q(a^{\mathrm{ref}} + h)$ be an exact solution of the corresponding non-linear action-space problem. Then

$$q(a^{\mathrm{ref}} + h_{\mathrm{FMQ}}) - q(a^{\mathrm{ref}}) \ge \rho_r \|g\|_2 - \tfrac{L}{2}\rho_r^2, \tag{32}$$

$$0 \le q(a^{\mathrm{ref}} + h^\star) - q(a^{\mathrm{ref}} + h_{\mathrm{FMQ}}) \le L\rho_r^2 = L(1-r)^2\eta^2. \tag{33}$$

In particular, the FMQ *target strictly improves the fixed critic whenever $0 < \rho_r < 2\|g\|_2/L$.*

Proof. By L-smoothness, for every $\|h\|_2 \le \rho_r$,

$$\left|q(a^{\mathrm{ref}} + h) - q(a^{\mathrm{ref}}) - \langle g, h\rangle\right| \le \frac{L}{2}\|h\|_2^2.$$

Evaluating the lower bound at h_{FMQ}, where $\langle g, h_{\mathrm{FMQ}}\rangle = \rho_r\|g\|_2$ and $\|h_{\mathrm{FMQ}}\|_2 = \rho_r$, gives eq. (32). Applying the upper bound to $h^\star$, using $\langle g, h^\star\rangle \le \rho_r\|g\|_2$, and subtracting the FMQ lower bound yields the $L\rho_r^2$ non-linear suboptimality bound in eq. (33); the lower bound 0 holds by optimality of $h^\star$. □

This is the standard linear-model Cauchy-step analysis from classical trust-region optimization [Nocedal and Wright, 2006, Xie et al., 2024], specialized to a flow-map endpoint. Thus, FMQ is exactly optimal for the linearized, per-sample trust-region subproblem and is within $O((1-r)^2\eta^2)$ of the exact non-linear action-space optimum under local critic smoothness.

Relation to policy-level trust regions. FMQ is not a KL-constrained TRPO update [Schulman et al., 2015] and does not inherit TRPO's KL-based guarantee. Instead, it applies the classical trust-region principle—optimizing a local model within a norm-bounded neighborhood—to the flow-map output [Nocedal and Wright, 2006, §4.1]. For fixed (s, r, a_r) and $g = \nabla_a Q_\phi(s, a^{\mathrm{ref}})$, a velocity update of radius η_{eff} induces the action displacement

$$h_{\mathrm{FMQ}} = (1-r)\,\eta_{\mathrm{eff}}\,\frac{g}{\|g\|_2} = \rho_r \frac{g}{\|g\|_2}, \qquad \rho_r := (1-r)\,\eta_{\mathrm{eff}}.$$

Hence, for $g \neq 0$, FMQ is exactly the linear-model Cauchy solution $h_{\mathrm{FMQ}} \in \arg\max_{\|h\|_2 \le \rho_r}\langle g, h\rangle$, consistent with normalized-gradient trust-region analyses [Xie et al., 2024, Cor. 1], and by proposition B.1 it is within $L\rho_r^2 = L(1-r)^2\eta_{\mathrm{eff}}^2$ of the corresponding non-linear action-space optimum if the fixed critic is L-smooth.

The map-output constraint also controls the policy-level Wasserstein distance. Writing $\pi^{\text{ref}}(\cdot \mid s) = T^{\text{ref}}(s,\cdot)_{\#}\nu_s$ and $\pi^{+}(\cdot \mid s) = T^{+}(s,\cdot)_{\#}\nu_s$, and coupling both policies through the same $z \sim \nu_s$, gives

$$\begin{aligned} W_2^2\big(\pi^{+}(\cdot \mid s), \pi^{\text{ref}}(\cdot \mid s)\big) &\le \mathbb{E}_{z\sim\nu_s}\|T^{+}(s,z) - T^{\text{ref}}(s,z)\|_2^2 \\ &= \mathbb{E}_{z\sim\nu_s}\big[\rho_r(s,z)^2\big] = \mathbb{E}_{z\sim\nu_s}\big[(1-r)^2\eta_{\text{eff}}(s,z)^2\big]\,. \end{aligned}$$

Because this shared-source coupling need not be optimal, the inequality is generally strict. The map-space constraint of FMQ is therefore a tractable, conservative sufficient condition for remaining within a Wasserstein neighborhood—not an exact Wasserstein policy optimization or a KL-TRPO update. Related Wasserstein stability bounds appear in [Albergo and Vanden-Eijnden, 2023, Boffi et al., 2024], and the geometry is related to Wasserstein policy trust regions [Terpin et al., 2022].

Proposition B.2 (Reference-policy improvement)**.** *Let $\pi^{\text{ref}}(\cdot \mid s) = T^{\text{ref}}(s,\cdot)_{\#}\nu_s$, assume $Q_\phi = Q^{\pi^{\text{ref}}}$, and suppose $Q^{\pi^{\text{ref}}}(s,\cdot)$ is L_s-smooth. Define*

$$g(s,z) = \nabla_a Q^{\pi^{\text{ref}}}\big(s, T^{\text{ref}}(s,z)\big)\,, \qquad \widehat{g}(s,z) = \begin{cases} g(s,z)/\|g(s,z)\|_2, & g(s,z) \neq 0, \\ 0, & g(s,z) = 0, \end{cases}$$

and $T^{+}(s,z) = T^{\text{ref}}(s,z) + \rho(s,z)\,\widehat{g}(s,z)$, $\pi^{+}(\cdot \mid s) = T^{+}(s,\cdot)_{\#}\nu_s$. If $T^{+}(s,z)$ is feasible and $0 \le \rho(s,z) \le 2\|g(s,z)\|_2/L_s$ for every (s,z), then

$$J(\pi^{+}) - J(\pi^{\text{ref}}) \ge \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^{\pi^{+}},\, z\sim\nu_s}\left[\rho(s,z)\|g(s,z)\|_2 - \frac{L_s}{2}\rho(s,z)^2\right] \ge 0,$$

where $d^{\pi^{+}}$ is the normalized discounted state-visitation distribution. Moreover, $V^{\pi^{+}}(s) \ge V^{\pi^{\text{ref}}}(s)$ for every s.

Proof. By L_s-smoothness,

$$Q^{\pi^{\text{ref}}}\big(s, T^{+}(s,z)\big) - Q^{\pi^{\text{ref}}}\big(s, T^{\text{ref}}(s,z)\big) \ge \rho(s,z)\|g(s,z)\|_2 - \frac{L_s}{2}\rho(s,z)^2 \ge 0.$$

Averaging over $z \sim \nu_s$ gives a non-negative expected advantage of π^{+} with respect to π^{ref} at every state. The performance-difference lemma [Kakade and Langford, 2002], applied also from each possible initial state, yields both claims. □

For FMQ, $\rho(s,z) = (1-r)\,\eta_{\text{eff}}(s,z)$. Proposition B.2 is a conditional guarantee assuming an exact reference-policy critic and exact realization of the FMQ target; it does not imply unconditional monotonic improvement for a learned critic or across online iterations.

Approximate critic. The practical critic is explicitly tied to the reference policy: although it is updated on the expanding online replay buffer, its Bellman target in eq. (8) uses the frozen offline flow map, so Q_ϕ is trained to evaluate π^{off} rather than the changing online actor, although finite data and function approximation make it inexact. The exactness assumption can also be relaxed. If the step uses an estimated gradient $\hat{g}$ in place of g, i.e., $h = \rho\,\hat{g}/\|\hat{g}\|_2$, the same smoothness argument gives

$$Q^{\pi^{\text{ref}}}(s, a^{\text{ref}} + h) - Q^{\pi^{\text{ref}}}(s, a^{\text{ref}}) \ge \rho\,\frac{\langle g, \hat{g}\rangle}{\|\hat{g}\|_2} - \frac{L_s}{2}\rho^2,$$

so an approximate critic still guarantees improvement when $\langle g, \hat{g}\rangle > 0$ and $\rho \le 2\langle g, \hat{g}\rangle/(L_s\|\hat{g}\|_2)$. The estimated gradient therefore need not be exact, but it must retain sufficient alignment with the reference-policy gradient, and the endpoint step must be sufficiently small. FMQ does not estimate L_s or this alignment, so twin-critic disagreement is a practical surrogate—not a formal certificate—for this condition: normalization bounds every ideal target, while the adaptive radius contracts it where the critics disagree. Section C.2 evaluates robustness when these conditions are violated.

C Additional Experiments

C.1 Comparison with the Originally Reported MVP Results

On the six tasks for which the original MVP* paper reports results, FMQ matches or exceeds MVP* on all six (five wins and one tie), with mean success 0.923 versus 0.838, an 8.5-point or 10.1% relative improvement; FMQ + QGBS reaches 0.938 (12.0% relative). We clearly distinguish this comparison from the larger 21.3% gain against our controlled MVP reproduction in section 4, which allows evaluating all 12 environments under identical conditions (table 4).

Table 4: Comparison with the results reported in the original MVP paper (MVP*) on the six shared tasks (success rate, from table 1).

Environment	MVP*	FMQ	FMQ + QGBS
`can`	0.92	0.96	0.97
`square`	0.93	0.94	0.95
`cube-dbl-t3`	1.00	1.00	1.00
`cube-dbl-t4`	0.95	0.98	1.00
`cube-trl-t3`	0.71	0.78	0.84
`cube-trl-t4`	0.52	0.88	0.87
Mean	0.838	0.923	0.938

C.2 Dependence on Critic

Critic accuracy is a central challenge in offline-to-online RL. FMQ addresses it by design as the trust region caps each update at η_{eff}, and the adaptive radius shrinks where the ensemble disagrees. Intuitively, twin-critic disagreement is a computationally cheap proxy for epistemic uncertainty. We therefore interpret the adaptive radius as a heuristic that contracts potentially unreliable updates, rather than as a formal safety certificate. To test this further we design two ablations.

Early online adaptation. We evaluate adaptation at 100k-step resolution. At the beginning of online adaptation the critic is at its worst and Q-values are the most biased. Mean success increases from 18% at the offline checkpoint to 61% at the first 100k online checkpoint, versus 38% for MVP and 58% for QC. Thus, we observe no sustained transition drop at this resolution. Table 5 tracks the progress of FMQ and baselines and highlights that the margin of FMQ over MVP peaks early, suggesting FMQ is more robust to critic error than the best-of-N baselines.

Table 5: Average success rate (%) across all 12 environments after K steps of online fine-tuning, following 1M gradient steps of offline pre-training ($K = 0$: offline pre-trained policy).

Method	$K = 0$	100k	200k	300k	500k	1M
FMQ	18	**61**	**72**	**77**	**85**	**91**
MVP	18	38	50	53	63	73
QC	**35**	58	61	65	73	82

FMQ with critic-gradient noise. We next inject noise into the critic gradient—effectively corrupting the unit-normed FMQ target. Here, a miscalibrated critic yields gradient directions that deviate from the true ascent direction on poorly-estimated states. We emulate this by injecting c std of Gaussian noise into every online policy update; FMQ is otherwise fine-tuned exactly as in the paper. Even at $c = 1.0$, where noise equals the true gradient in magnitude, no task collapses and average success declines only $0.62 \rightarrow 0.54$ on the six hardest tasks (table 6). As the critic target is unit normalized, a noise level of $c = 1$ is effectively a large noise injection. We observe that despite such considerable amounts of noise the FMQ performance deteriorates by a proportionally smaller amount, with the greatest relative drop of 24.5% in the `cube-trl-t4` environment. This highlights that FMQ remains relatively robust to critic gradient noise.

Table 6: Critic-gradient corruption experiment. FMQ fine-tuned exactly as in the paper; IQM with 95% bootstrap CI pooled over 6 tasks $\times$ 2 seeds.

Environment	$c = 0.0$	$c = 0.5$	$c = 1.0$
`cube-trl-t3`	0.63 ± 0.03	0.61 ± 0.05	0.60 ± 0.07
`cube-trl-t4`	0.61 ± 0.03	0.51 ± 0.01	0.46 ± 0.02
`hmaze-med-t3`	0.72 ± 0.05	0.66 ± 0.04	0.57 ± 0.01
`hmaze-med-t4`	0.10 ± 0.04	0.06 ± 0.00	0.07 ± 0.01
`amaze-gnt-t4`	0.77 ± 0.04	0.72 ± 0.04	0.65 ± 0.09
`amaze-gnt-t5`	0.87 ± 0.05	0.87 ± 0.03	0.86 ± 0.00
Average	**0.62**	**0.57**	**0.54**

C.3 Efficiency Analysis with Wall Clock Time

FMQ is broken up into two distinct phases like all offline-to-online algorithms: (1) training of the flow map policy and (2) evaluation. Consequently, we position FMQ as a training-time method, which leads to significantly faster and better training, while the evaluation stage is shared with other one-step baseline methods like MVP—i.e., the exact same experimental setup is used at evaluation time amongst methods.

Training. Per online update, FMQ computes a single closed-form Q-gradient target in one step, whereas MVP and QC construct targets via best-of-32 generate-and-select, using 1 and 10 ODE steps per candidate, respectively. In terms of critic evaluations, best-of-N selection requires N critic calls per decision for all methods, while the FMQ target adds one $\nabla_a Q$ computation per training update. To quantify the training speed advantage, we report the raw wall clock time of each method during the online phase with respect to achieving a percentage of its final performance (table 7). Clearly, FMQ converges faster to its final performance in comparison to MVP and QC baselines. Specifically, the total online run time of FMQ (149 min) matches that of MVP (151 min; $1.01\times$ per-step ratio) and is $1.54\times$ lower than that of QC (229 min), confirming the closed-form Q-gradient target adds no per-step wall clock overhead.

We further analyze how fast each method converges. Following section 4.2, we define ξ as the highest success rate reachable by both FMQ and the baseline, computed per seed and environment, and measure the wall-clock time T to first reach $\{75\%, 85\%, 95\%, 100\%\}$ of ξ. The ratio $T_{\text{baseline}}/T_{\text{FMQ}}$, averaged over 5 seeds, quantifies how many times faster FMQ converges (table 8). We find that FMQ converges substantially faster: it reaches every performance milestone 2.1–$3.0\times$ sooner in wall-clock time than both baselines. FMQ requires only $N{=}1$ where best-of-N target construction needs $N{=}32$, yet converges 2–$3\times$ faster at matching total run time—demonstrating the efficiency of the closed-form target.

Table 7: Wall-clock time (min) during online training (1M steps) to reach a percentage of each method's final performance, averaged across all tasks.

Method	@75%	@85%	@95%	@100%	Total
MVP	74	90	114	124	151
QC	60	73	87	94	229
FMQ (ours)	**26**	**32**	**38**	**45**	**149**

Table 8: The ratio $T_{\text{baseline}}/T_{\text{FMQ}}$, averaged over 5 seeds, quantifies how many times faster FMQ converges.

	@75%	@85%	@95%	@100%
vs. MVP	$2.85\times$	$2.81\times$	$3.00\times$	$2.76\times$
vs. QC	$2.31\times$	$2.28\times$	$2.29\times$	$2.09\times$

Inference. The evaluation protocol is inherited and shared: best-of-32 is used by all methods to ensure fair comparison. To measure the impact of the closed-form Q-gradient update at inference, we also evaluate all methods at $N \in \{1, 8\}$, including QGBS ($B{=}N$, $K{=}1$, $M{=}1$), which applies the FMQ trust-region update to the candidates before selecting the highest Q-value action, and measure the raw wall-clock latency (ms) per action (table 9). We observe that: (1) FMQ and MVP have identical inference cost—both are one-step policies with best-of-N selection (0.23 vs. 0.22 ms at $N{=}1$)—so the accuracy gains of FMQ (40.9 vs. 24.3 at $N{=}1$; 92.7 vs. 72.5 at $N{=}8$) come at zero additional latency. (2) QGBS applies the trust-region update before selection, at $\sim 1.6\times$ the latency of FMQ, and yields large gains at $N{=}1$ (76.1 vs. 40.9). This advantage diminishes as more candidates are sampled (93.0 vs. 92.7 at $N{=}8$), since best-of-N recovers high Q-value actions. (3) Even with this overhead, FMQ + QGBS is faster than QC at every budget (0.37 vs. 0.58 ms at $N{=}1$), since QC requires 10 Euler ODE steps per candidate. Since all N candidates are processed as a single batch through policy and critic, latency grows sublinearly in N ($0.23 \rightarrow 0.31$ ms from $N{=}1$ to 8). We therefore conclude that FMQ at inference matches the latency of MVP, since both use best-of-N, and that the trust-region update—whose cost is negligible at training (149 vs. 151 min above)—can optionally be applied at inference (QGBS), where its modest overhead substantially improves performance when the candidate budget is small.

C.4 Performance Analysis of QGBS

QGBS degrades performance on `hmaze-med-t3` and `amaze-gnt-t4` (table 1). We attribute this to task-level sensitivity to critic error, amplified by the reliance of QGBS on the critic: it ranks and selects among $M \cdot B$ candidates at every beam step. Our corruption experiment (section C.2)

Table 9: Success rate (%, IQM over 12 environments) with per-action latency in parentheses (ms).

Method	$N = 1$	$N = 8$
FMQ + QGBS (B=N, K=1, M=1)	**76.1** (0.37)	**93.0** (0.45)
FMQ (BoN)	40.9 (0.23)	92.7 (0.31)
QC	43.0 (0.58)	82.7 (0.67)
MVP	24.3 (0.22)	72.5 (0.31)

substantiates this: the three most noise-sensitive tasks are the three where QGBS does not improve FMQ (table 10). Importantly, the core results of FMQ (table 1) do not depend on QGBS: FMQ already achieves state-of-the-art performance, with QGBS as an optional inference-time refinement.

We consider the following approaches to decide when QGBS should be disabled: (1) Since QGBS is purely inference-time, comparing FMQ vs. FMQ + QGBS on a few evaluation episodes costs no training and settles the choice per task. (2) A priori, the critic-ensemble disagreement $\tilde{\delta}_{\text{critic}}$ (eq. (13)) measures exactly what QGBS depends on—whether the critic can rank diverse candidates; high end-of-training disagreement signals QGBS should be disabled in favor of best-of-N, consistent with the sensitivity pattern in table 10. Notably, this gating reuses the same uncertainty signal that drives the adaptive trust region of FMQ without additional cost.

Table 10: Performance drop due to c-level critic gradient noise (table 6) versus the effect of QGBS on FMQ (table 1). `ctl` = `cube-trl`, `hm` = `hmaze-med`, `ag` = `amaze-gnt`.

	`ctl-t3`	`ctl-t4`	`hm-t3`	`hm-t4`	`ag-t4`	`ag-t5`
Noise sensitivity (c: $0 \to 1.0$)	-0.03	$\mathbf{-0.15}$	$\mathbf{-0.15}$	-0.03	$\mathbf{-0.12}$	-0.01
QGBS effect (table 1)	$+0.06$	-0.01	$\mathbf{-0.11}$	0.00	-0.03	0.00

C.5 Ablations of the Trust Region

Trust-region radius and critic disagreement coefficient. Trust-region radius and disagreement coefficient are not independent knobs: the effective radius η_{eff} is the base radius η scaled down per sample by critic disagreement, with β controlling the scaling strength (eq. (13)). Sweeping β therefore probes both questions at once: it varies the coefficient directly and, through the coupling, spans effective radii from near the full base radius ($\beta = 0.05$, $\eta_{\text{eff}} \approx \eta$) to a heavily contracted one ($\beta = 1.0$). Across this range, all settings fall within overlapping standard deviations on both tasks (table 11), and the adaptive rule matches or exceeds a fixed radius at the same base value. On these two tasks, performance is insensitive to β over the tested range.

Table 11: Ablation on β: success rate (%) for a fixed and an adaptive trust region.

Trust region	`cube-trl-t3`	`hmaze-med-t3`
Fixed, $\eta = 0.3$	80 ± 6	58 ± 9
Adaptive, $\beta = 0.05$	88 ± 4	58 ± 5
Adaptive, $\beta = 0.1$	86 ± 8	58 ± 9
Adaptive, $\beta = 0.3$ (default)	84 ± 7	66 ± 11
Adaptive, $\beta = 0.5$	83 ± 6	62 ± 6
Adaptive, $\beta = 1.0$	88 ± 7	56 ± 4

Normalized vs. unnormalized Q-gradients. With the raw gradient g, the effective step becomes $\eta\|g\|_2$, which varies substantially across states and grows as the critic trains online. Empirically, unnormalized guidance collapses to 0% success. This is consistent with theorem 3.2: the optimal trust-region update *is* the normalized gradient scaled by η, so normalization enforces the constraint rather than serving as a heuristic.

Critic choice and online data-collection policy. All methods (QC, MVP, FMQ) share the same critic architecture and training (clipped double Q-learning) and the same data-collection protocol, following the standard practice in this line of work [Park et al., 2025b], ensuring fair evaluation.

D Successful Rollouts

We visualize successful rollouts from the trained FMQ policy across all 12 evaluation environments. Each figure shows uniformly-spaced frames from a single episode that achieves the task goal.

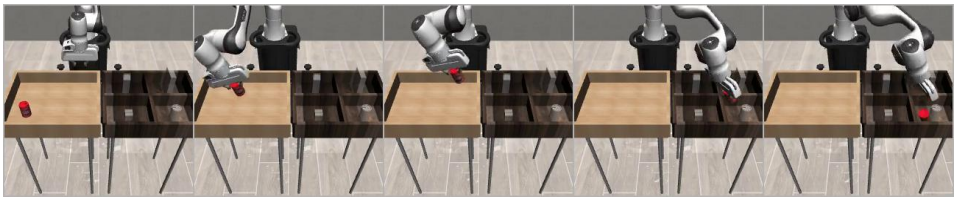


Figure 5: `can` (Robomimic). Pick up a can from the table and place it into the bin.



Figure 6: `square` (Robomimic). Pick up a square nut and fit it onto a peg.

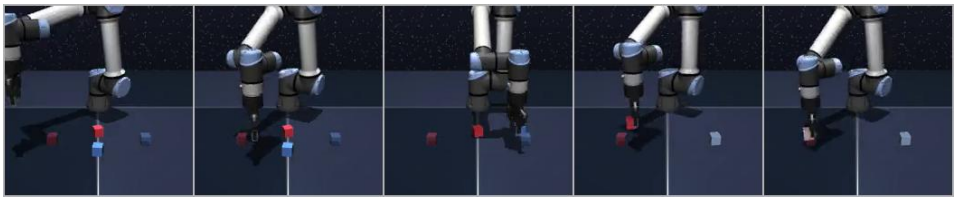


Figure 7: `cube-double-task3` (OGBench). Rearrange 2 cubes to target positions.

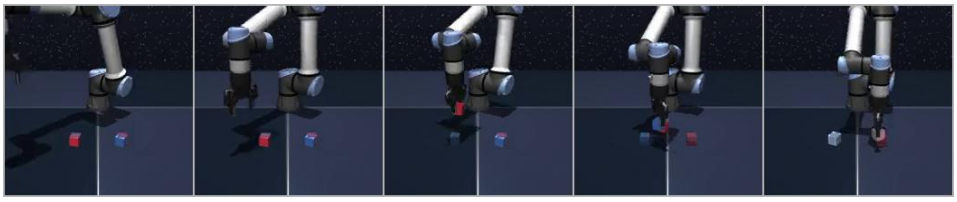


Figure 8: `cube-double-task4` (OGBench). Swap the positions of 2 cubes.

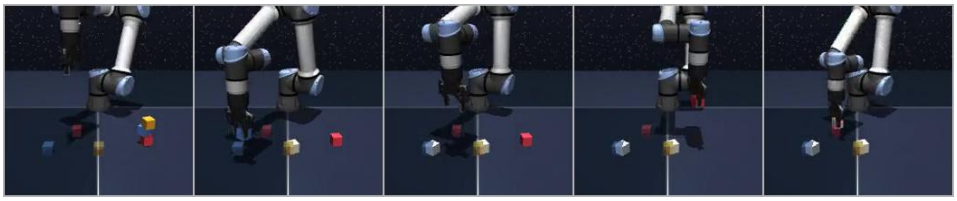


Figure 9: `cube-triple-task3` (OGBench). Unstack 3 cubes and place them at separate target positions.

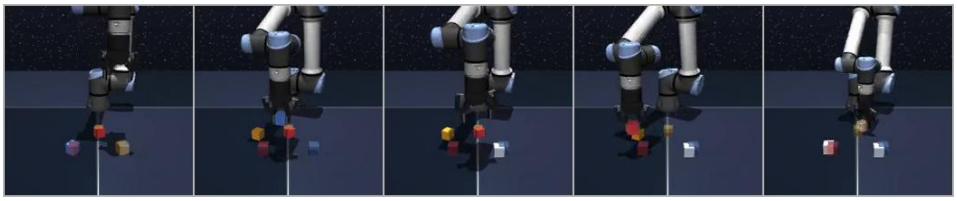


Figure 10: `cube-triple-task4` (OGBench). Cyclically permute 3 cubes to new positions.

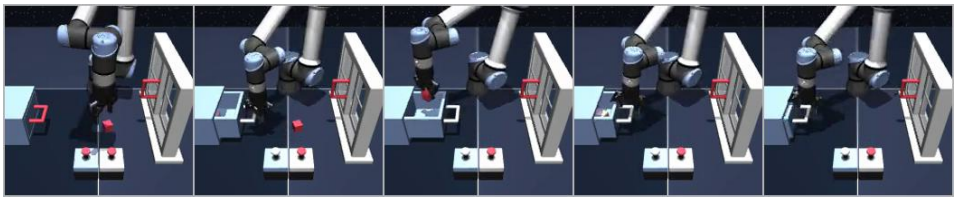


Figure 11: `scene-task4` (OGBench). Unlock the drawer button, open the drawer, and place the cube inside.

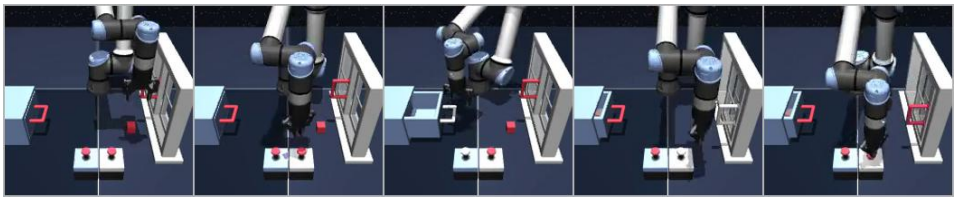


Figure 12: `scene-task5` (OGBench). Place the cube in the drawer and open the window.



Figure 13: `humanoidmaze-medium-task3` (OGBench). Navigate a humanoid to a goal in a medium maze.



Figure 14: `humanoidmaze-medium-task4` (OGBench). Navigate a humanoid to a distant goal in a medium maze.

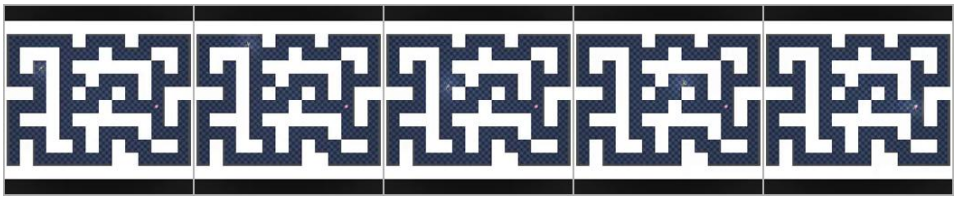


Figure 15: `antmaze-giant-task4` (OGBench). Navigate an ant to a goal across a giant maze.

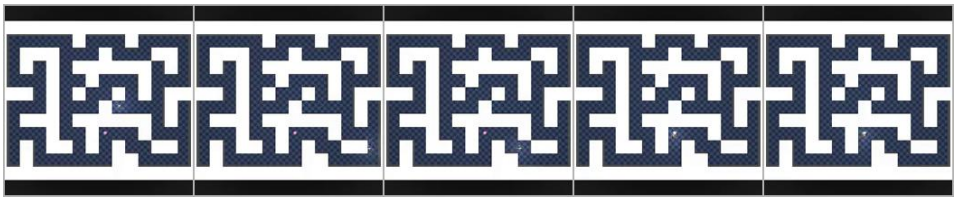


Figure 16: `antmaze-giant-task5` (OGBench). Navigate an ant to a nearby goal in a giant maze.

E Algorithms

Algorithm 1 FLOW MAP Q-GUIDANCE (FMQ)

Require: Offline policy $u^{\text{off}}_{r,1}$, online policy $u^{\theta}_{r,1}$, critics Q_{ϕ_1}, Q_{ϕ_2}, buffer $\mathcal{D}$

for each environment step **do**
  $a_1 \leftarrow a_0 + u^{\theta}_{0,1}(a_0|s)$, $a_0 \sim \mathcal{N}(0, I)$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s, a_1, r, s')\}$
  Sample batch from $\mathcal{D}$; update critics via Eq. 8
  $r \sim \mathcal{U}[0, 1)$; $a_0 \sim \mathcal{N}(0, I)$; $a_r \leftarrow (1-r)a_0 + r\, a_{\text{data}}$
  $a_1 \leftarrow a_r + (1-r)\, u^{\text{off}}_{r,1}(a_r|s)$
  $g \leftarrow \nabla_a Q_{\phi_1}(s, a_1)/(\|\nabla_a Q_{\phi_1}(s, a_1)\|_2 + \kappa_1)$
  $\eta_{\text{eff}} \leftarrow \eta/(1 + \beta\, \tilde{\delta}_{\text{critic}})$ ▷ Eq. 13
  $\theta \leftarrow \theta - \alpha\, \nabla_\theta \|u^{\theta}_{r,1}(a_r|s) - \text{sg}(u^{\text{off}}_{r,1}(a_r|s) + \eta_{\text{eff}}\, g)\|^2$
end for

Algorithm 2 Q-GUIDED BEAM SEARCH (QGBS)

Require: Flow map $X^{\theta}_{r,1}$, critic Q_ϕ, state s, beam M, steps K, branches B, SNR ρ, step size η

$t' \leftarrow \rho/(1+\rho)$
Sample $\{a^m_0\}_{m=1}^M \sim \mathcal{N}(0, I)$; $a^m_1 \leftarrow a^m_0 + u^{\theta}_{0,1}(a^m_0|s)$ for all m
for $k = 1, \ldots, K$ **do**
  for $m = 1, \ldots, M$ and $b = 1, \ldots, B$ **do**
    $\varepsilon^{mb} \sim \mathcal{N}(0, I)$
    $\hat{a}^{mb}_1 \leftarrow X^{\theta}_{t',1}\big(t'\, a^m_1 + (1-t')\, \varepsilon^{mb} \mid s\big)$ ▷ Re-noise & complete
  end for
  $\{a^m_1\}_{m=1}^M \leftarrow \text{Top-}M\big(\{\hat{a}^{mb}_1\}_{m,b};\ Q_\phi(s, \hat{a}^{mb}_1)\big)$ ▷ Select best M of $M{\cdot}B$
  $a^m_1 \leftarrow a^m_1 + \eta\, \nabla_a Q_\phi(s, a^m_1)/\|\nabla_a Q_\phi(s, a^m_1)\|_2$ for all m ▷ Thm. 3.2
end for
return $a_1^{\arg\max_m Q_\phi(s, a^m_1)}$

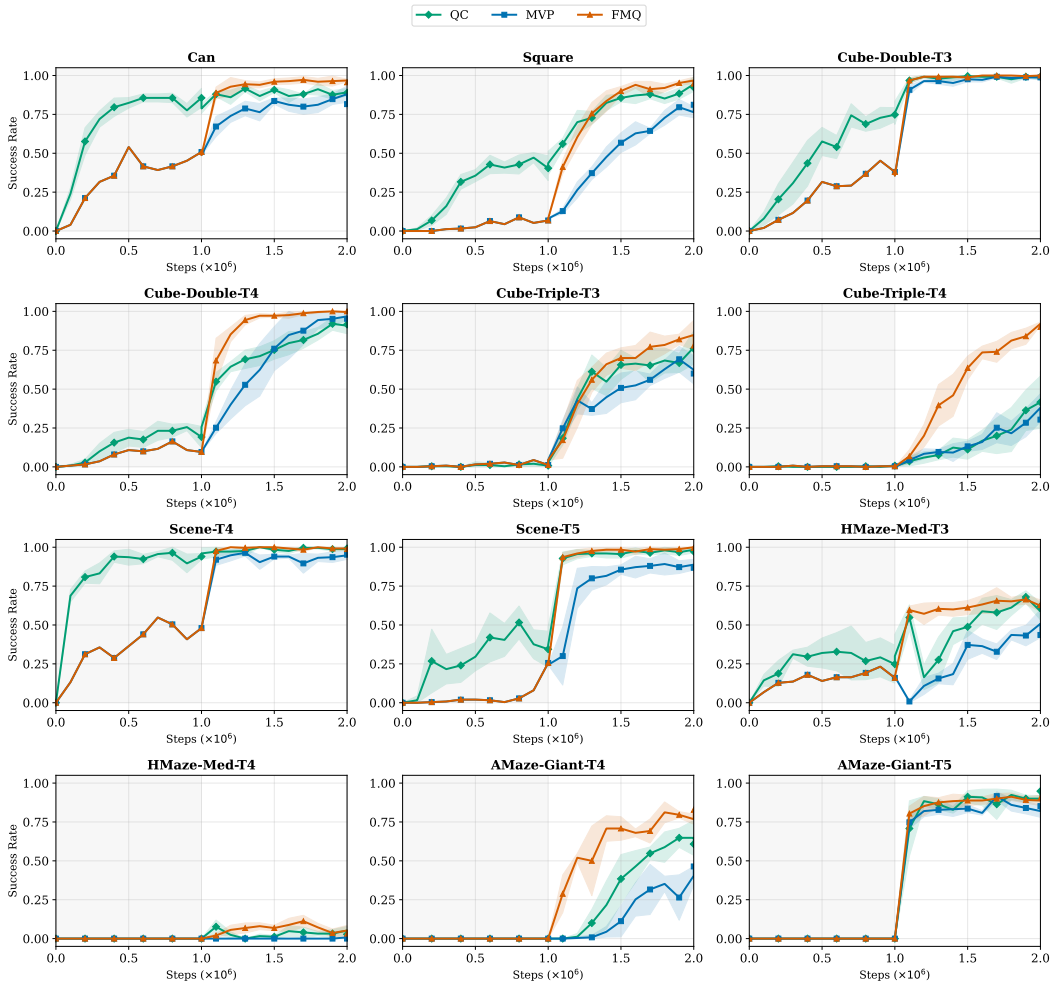


Figure 17: Offline-to-online learning curves for QC, MVP, and FMQ on all environments. All methods perform 1M offline followed by 1M online steps. Shaded regions indicate 95% CIs over 5 seeds.

F Training Curves

Figure 17 extends fig. 2 to all 12 environments. On the simpler manipulation tasks (`can`, `square`, `cube-dbl`), all methods converge to near-perfect success, but FMQ reaches this level earlier. The advantage becomes more pronounced on the harder tasks: on `cube-trl-t4`, FMQ reaches 0.88 while MVP plateaus at 0.32; on `amaze-gnt-t4`, FMQ achieves 0.80 versus 0.42 for MVP. For locomotion (`hmaze`, `amaze`), the Q-gradient signal is particularly beneficial under sparse rewards, where best-of-N selection provides a weaker learning signal.

G Trust-Region Convergence

Figure 18 extends the convergence analysis of fig. 4 to all 12 environments. We track the action displacement $\|u^{\text{on}}_{r,1} - u^{\text{off}}_{r,1}\|_2$ between the online and frozen offline flow map policies throughout online training. At the onset of fine-tuning (1M steps), both policies coincide and the displacement is near zero. As training progresses, the trust-region loss in eq. (12) drives $u^{\text{on}}_{r,1}$ toward $u^{\text{off}}_{r,1} + \eta_{\text{eff}}\,\hat{g}$, causing the displacement to grow monotonically until it stabilizes near η_{eff}. The orange curve (right axis) shows the implied Q-uncertainty $\tilde{\sigma}_Q = (\eta/\eta_{\text{eff}} - 1)/\beta$, which decreases as the critic

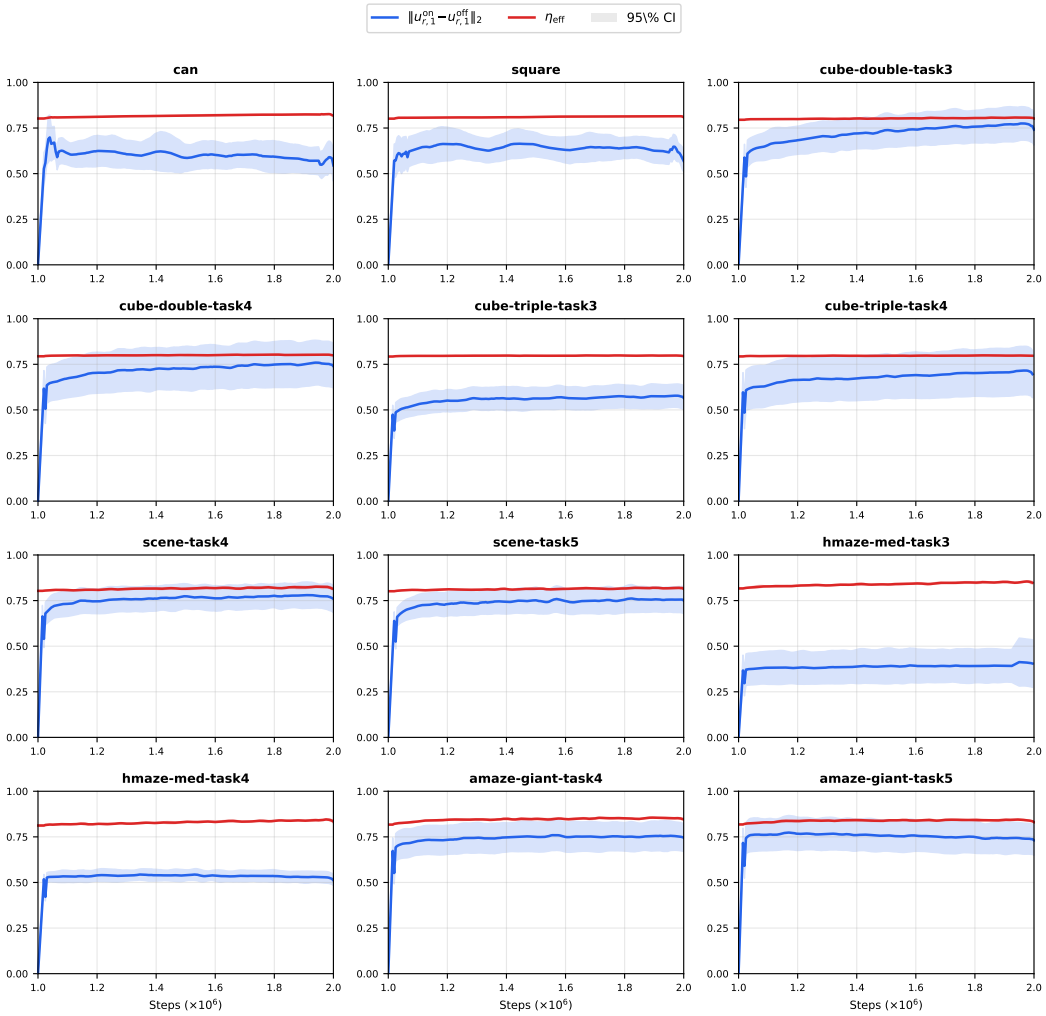


Figure 18: Trust-region convergence for FMQ (β=0.3) across all 12 environments. Blue: action displacement $\|u^{\text{on}}_{r,1} - u^{\text{off}}_{r,1}\|_2$. Red dashed: adaptive trust-region radius η_{eff}. Orange dotted (right axis): implied Q-uncertainty $\tilde{\sigma}_Q = (\eta/\eta_{\text{eff}} - 1)/\beta$.

becomes more confident—automatically tightening the trust region and confirming that the adaptive mechanism prevents overshooting in low-confidence regions.

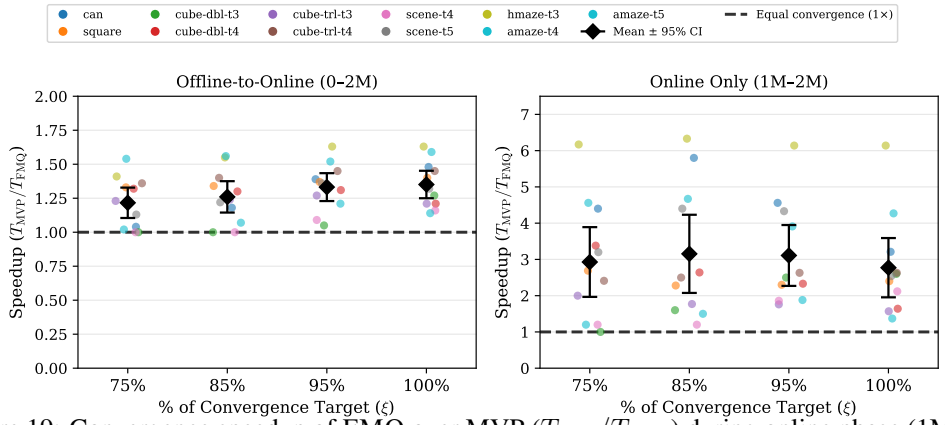


Figure 19: Convergence speedup of FMQ over MVP ($T_{\text{MVP}}/T_{\text{FMQ}}$) during online phase (1M–2M steps). Each dot represents one environment; black diamonds show the mean with 95% CI. The dashed line marks equal convergence speed ($1\times$).

Table 12: Speedup of FMQ over MVP ($T_{\text{MVP}}/T_{\text{FMQ}}$) measuring time to reach a fraction of the convergence target ξ (per seed, averaged over 5 seeds). **Left**: full training (0–2M). **Right**: online phase only (1M–2M). Values > 1 indicate FMQ is faster.

	Full (0–2M)				**Online (1M–2M)**			
Environment	75%	85%	95%	100%	75%	85%	95%	100%
`can`	1.04	1.18	1.39	1.48	4.40	5.80	4.56	3.21
`square`	1.33	1.34	1.37	1.40	2.69	2.28	2.30	2.40
`cube-double-task3`	1.00	1.00	1.05	1.27	1.00	1.60	2.50	2.60
`cube-double-task4`	1.32	1.30	1.31	1.21	3.38	2.64	2.33	1.64
`cube-triple-task3`	1.23	1.24	1.27	1.21	2.00	1.77	1.76	1.57
`cube-triple-task4`	1.36	1.40	1.45	1.45	2.41	2.50	2.63	2.63
`scene-task4`	1.00	1.00	1.09	1.16	1.20	1.20	1.86	2.12
`scene-task5`	1.13	1.22	1.36	1.32	3.20	4.40	4.33	2.54
`hmaze-med-task3`	1.41	1.55	1.63	1.63	6.17	6.33	6.14	6.14
`amaze-giant-task4`	1.54	1.56	1.52	1.59	4.56	4.67	3.91	4.27
`amaze-giant-task5`	1.02	1.07	1.21	1.14	1.20	1.50	1.88	1.37
Average	**1.22**	**1.26**	**1.33**	**1.35**	**2.93**	**3.15**	**3.11**	**2.77**
95% CI	[1.10, 1.33]	[1.14, 1.37]	[1.23, 1.43]	[1.25, 1.45]	[1.97, 3.89]	[2.08, 4.23]	[2.27, 3.95]	[1.95, 3.59]

H Speedup Analysis

Figure 19 visualizes the per-environment convergence speedup of FMQ over MVP during the online phase (1M–2M), complementing the discussion in section 4.2. For each threshold $\xi \in 75\%, 85\%, 95\%, 100\%$ of the shared convergence target, we plot the ratio $T_{\text{MVP}}/T_{\text{FMQ}}$. FMQ is faster than MVP on every environment at every threshold (all points above $1\times$), with average speedups of 2.8–3.2×. The full per-environment breakdown including both full-training and online-only phases is provided in table 12.

Table 13: QGBS on FMQ. SNR= 1.5, $\eta = 0.3$. Columns grouped by K; sub-columns $\{B, M\}$. Success rate (mean $\pm$ std, 5 seeds, 50 eps). Best per row in **bold**.

Environment	$K=0$	$K=1$				$K=2$	
	$\{1,32\}$	$\{1,16\}$	$\{2,8\}$	$\{4,4\}$	$\{4,16\}$	$\{1,16\}$	$\{4,4\}$
can	0.96 ± 0.04	0.97 ± 0.02	0.96 ± 0.04	0.97 ± 0.03	0.94 ± 0.04	0.95 ± 0.03	$\mathbf{0.98 \pm 0.03}$
square	0.94 ± 0.02	0.94 ± 0.03	$\mathbf{0.96 \pm 0.04}$	0.95 ± 0.04	$\mathbf{0.96 \pm 0.02}$	0.94 ± 0.04	0.94 ± 0.03
cdp3	$\mathbf{1.00 \pm 0.00}$	0.99 ± 0.01	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
cdp4	0.98 ± 0.02	0.99 ± 0.03	0.99 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.98 ± 0.02	0.99 ± 0.03	0.99 ± 0.01
ctrp3	0.78 ± 0.10	0.82 ± 0.10	0.78 ± 0.06	$\mathbf{0.84 \pm 0.04}$	0.83 ± 0.07	$\mathbf{0.84 \pm 0.08}$	0.82 ± 0.08
ctrp4	$\mathbf{0.88 \pm 0.07}$	0.84 ± 0.06	$\mathbf{0.88 \pm 0.06}$	0.87 ± 0.05	0.82 ± 0.09	0.82 ± 0.04	0.84 ± 0.06
sc4	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	0.99 ± 0.01	0.99 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.99 ± 0.01	$\mathbf{1.00 \pm 0.00}$
sc5	0.98 ± 0.02	$\mathbf{1.00 \pm 0.00}$	0.99 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.99 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.98 ± 0.02
hm3	0.69 ± 0.04	0.70 ± 0.07	0.63 ± 0.04	0.58 ± 0.07	$\mathbf{0.72 \pm 0.11}$	0.68 ± 0.10	0.70 ± 0.07
hm4	0.06 ± 0.03	0.07 ± 0.04	0.10 ± 0.04	0.06 ± 0.03	$\mathbf{0.11 \pm 0.04}$	0.10 ± 0.05	0.10 ± 0.03
ag4	0.80 ± 0.06	0.78 ± 0.06	$\mathbf{0.86 \pm 0.10}$	0.77 ± 0.03	0.82 ± 0.04	0.75 ± 0.05	0.79 ± 0.03
ag5	$\mathbf{0.92 \pm 0.04}$	$\mathbf{0.92 \pm 0.04}$	0.90 ± 0.05	$\mathbf{0.92 \pm 0.05}$	0.90 ± 0.09	0.89 ± 0.03	0.90 ± 0.02
IQM	$0.91\ [0.89, 0.93]$	$0.92\ [0.90, 0.93]$	$\mathbf{0.93}\ [\mathbf{0.91}, \mathbf{0.95}]$	$\mathbf{0.93}\ [\mathbf{0.91}, \mathbf{0.94}]$	$0.92\ [0.90, 0.93]$	$0.90\ [0.89, 0.92]$	$0.91\ [0.90, 0.93]$

I Inference-Time Beam Search

Table 13 provides the full per-environment breakdown of QGBS applied to the trained FMQ checkpoint, extending the aggregate IQM results reported in table 2. NFE $= M(1 + KB)$, where M is the number of initial candidates, K the number of renoising steps, and B the number of completions per candidate; K=0 reduces to standard best-of-M. The per-environment results confirm that the gains from renoising (K=1) are consistent across task domains—manipulation, multi-object rearrangement, and locomotion—with the most notable improvements on the harder maze tasks (hm3: $0.59 \rightarrow 0.72$, hm4: $0.07 \rightarrow 0.11$).

Table 14: Hyperparameters shared across all methods.

Parameter	**Value**
Optimizer	Adam
Learning rate	3×10^{-4}
Batch size	256
Discount (γ)	0.99
Target update (τ)	5×10^{-3}
UTD ratio	1
Offline / online steps	1M / 1M
Replay buffer	2M
Policy network	MLP, 4×512, GELU
Critic network	MLP, 4×512, GELU, LayerNorm
Critic ensemble	2 (double Q, mean agg.)
Fourier embedding	64 dim per time axis
Chunking horizon (H)	5
Eval interval / episodes	100K / 50

J Implementation Details

All methods share the same network architecture, critic algorithm, and training pipeline to ensure a controlled comparison. The policy is parameterized as a time-conditioned velocity field u_θ: a 4-layer MLP with 512 hidden units and GELU activations. Scalar flow times are lifted to 64-dimensional sinusoidal Fourier embeddings before concatenation with the observation and noisy action. The critic follows the clipped double Q-learning framework [Fujimoto et al., 2018]: an ensemble of two Q-networks with the same MLP architecture (with LayerNorm) trained against a shared Bellman target using Polyak-averaged target networks (τ=0.005). All methods use action chunking (H=5), a replay buffer of 2M transitions, and are trained for 1M offline followed by 1M online steps with UTD ratio 1, Adam (lr=3×10^{-4}), and batch size 256. Full shared hyperparameters are in table 14.

QC [Li et al., 2025] trains a standard CFM velocity field $v_\theta(a_t, t \mid s)$ with the straight-line interpolation objective. At inference, the ODE is integrated from t=0 to t=1 with 10 Euler steps, producing 32 candidates scored by the critic (best-of-N, 320 NFE total).MVP [Zhan et al., 2026] replaces multi-

Table 15: Inference procedures. NFE = network forward evaluations per action.

Method	**Action selection**	**Steps**	N	**NFE**
QC	Best-of-N (Euler)	10	32	320
MVP	Best-of-N (flow map)	1	32	32
MVP + QGBS (ours)	QGBS	K	M	$K \cdot M$
FMQ (ours)	Best-of-N (flow map)	1	32	32
FMQ + QGBS (ours)	QGBS	K	M	$K \cdot M$

step Euler integration with a single-step flow map (K=1) that directly predicts the average velocity $u_{r,t}(a_r \mid s)$ over $[r, t]$. The network takes as input $[\mathbf{s}, \mathbf{x}_r, \text{Fourier}(r), \text{Fourier}(t), \text{Fourier}(t_c), \mathbf{a}_c]$ where $(t_c, \mathbf{a}_c)$ form a conditioning axis for stochastic action generation. Training uses a progressive curriculum: diagonal-only CFM (r=t) for 5K steps, then the interval $[r, t]$ is annealed to the full range over 50K steps, and the conditioning axis is introduced after 10K steps with $P(t_c{=}0){=}0.5$. At inference, a single forward pass generates each candidate and best-of-32 selection is applied (32 NFE). FMQ shares the same offline pretraining as MVP. During the online phase, it switches to the trust-region Q-gradient objective described in section 3.3: the offline flow map is frozen as the reference $u^{\text{off}}_{r,1}$, the Q-gradient is ℓ_2-normalized, and the trust-region radius η_{eff} adapts per sample via Q-ensemble disagreement (β=0.3, cf. eq. (13)). At inference, FMQ uses the same best-of-32 flow map selection as MVP (32 NFE). QGBS applies the Q-guided beam search of section 3.4 at inference time without additional training. Starting from a trained flow map, actions are diversified via SNR-based renoising and refined over K beam steps using M candidates, with the trust-region projection applied at each iteration. Steering strength is controlled by λ and actions are clipped via a straight-through estimator. Cost: $M(1 + KB)$ NFE per action. Inference procedures and computational costs are summarized in table 15. All training experiments were run on NVIDIA A100-SXM4-80GB GPUs. A full training run takes approximately 4 hours per seed. Inference-time steering evaluations were conducted on NVIDIA RTX 6000 Ada Generation GPUs (48 GB VRAM).

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our three primary contributions: the formalization of flow map policies, the derivation of the FLOW MAP Q-GUIDANCE (FMQ) algorithm and Q-GUIDED BEAM SEARCH (QGBS), and empirical analysis across 12 robotic manipulation and locomotion tasks from OGBench and RoboMimic.

   Guidelines:

   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: Limitations are explicitly discussed in Section 6 (Conclusion). We note that while FMQ enjoys an efficient linearized critic approximation, its effectiveness is inherently dependent on critic accuracy and local linearity. We also note that deployment on physical robotic platforms and curvature-aware updates remain open challenges for future work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The paper formally states Theorem 3.2 in Section 3.3, which details the optimal average velocity under a trust-region constraint. A proof sketch is provided immediately following the theorem in the main text, and the complete formal proof—including all necessary KKT conditions and assumptions (e.g., non-zero Q-gradient, Slater's condition)—is detailed in Appendix A.1. Additionally, Appendix A.2 provides a formal mathematical derivation demonstrating the equivalence of Eulerian and Mean Flow policies.

   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: Section 4 and Appendix G explicitly detail the experimental setup. We specify the training datasets (OGBench, RoboMimic), network architectures (4-layer MLP with 512 hidden units, Fourier embeddings), optimizer (Adam with 3×10^{-4} learning rate), batch size (256), and training durations (1M offline steps, 1M online steps). The clipped double Q-learning setup and algorithmic inference procedures (QGBS, best-of-N) are detailed in Appendix G and Table 7.

   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: We provide an anonymized code archive in the supplementary material containing training code, evaluation scripts, and the environment configurations needed to reproduce the results. All evaluation datasets and benchmarks (OGBench, RoboMimic) are publicly available. Upon publication, all codebase assets and model checkpoints will be released under a permissive open-source license.

   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: Appendix G (Table 6) comprehensively lists all hyperparameters. Evaluation details and inference budgets are detailed in Section 4 and Table 7.

   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: Yes. As detailed in Section 4 (Evaluation Protocol), we evaluate models across 5 independent random seeds and 50 episodes per environment. We report the Interquartile Mean (IQM) alongside 95% stratified bootstrap confidence intervals, following the rigorous RL evaluation protocols established by Agarwal et al. (2021). All learning curves (Figure 2, 17) and convergence speedup plots (Figure 3, 19) display 95% confidence intervals.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix G details the hardware and compute time required to reproduce our results.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: The work evaluates RL algorithms entirely within simulated public benchmark environments (OGBench, RoboMimic). It involves no human subjects, does not use personally identifiable information, and releases no inherently harmful models. The authors have reviewed and conform to 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: [Yes]

Justification: Our work advances sample-efficient reinforcement learning for robotics. Positive impacts include enabling robots to quickly adapt to new, dynamic environments, which could assist humans in dangerous or repetitive tasks. A potential negative societal impact is the broader automation of physical labor, which carries economic and workforce displacement risks. Additionally, deploying generative policies on physical hardware carries the risk of unpredictable, out-of-distribution physical actions if the critic is poorly calibrated.

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: This paper presents foundational algorithmic research in reinforcement learning and evaluates on standard simulated robotic benchmarks. We do not release models or datasets that inherently pose a high risk for dual-use or malicious misuse.

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 third-party assets and datasets, including OGBench [**?**], RoboMimic [**?**], and baseline methods such as MVP [Zhan et al., 2026] and QC [Li et al., 2025], are properly cited. We respect the open-source licenses (e.g., MIT, Apache 2.0) of the respective codebase frameworks used to develop our environments and neural network architectures.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The new assets consist of our training and evaluation codebase for FMQ and QGBS. These are provided in the supplementary material alongside a comprehensive README detailing environment installation, exact execution commands, and expected outputs.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research relies entirely on simulated physics environments and pre-existing offline datasets. No human subjects or crowdsourcing were involved.

    Guidelines:

    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 does not involve research with human subjects, so IRB approval is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology, training, or evaluation of FMQ. Any LLM use was limited to writing, editing, or formatting assistance and does not impact the methodology, scientific rigor, or originality of the research.

    Guidelines:

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