Q-Guided Flow Q-Learning

Abstract: Generative policies improve expressivity over Gaussian actors but often come with entangled training pipelines (e.g., joint actor–critic training, student–teacher distillation, or sequence-to-sequence planners). We introduce *Q-Guided Flow Q-Learning (QFQL)*, an actor–critic framework where the actor is trained *independently* via conditional flow matching for behavior cloning, and the critic is trained *independently* via temporal-difference (TD) learning. At inference, actions are produced by integrating the learned flow field and adding a value-seeking correction proportional to the action-gradient of the critic, i.e., a guidance term $\beta \nabla_a Q(s, a)$. This decoupled design simplifies optimization, reduces instability from joint updates, and enables controllable trade-offs between behavior realism and value seeking at test time. Empirically, QFQL achieves strong offline RL performance and stable training across tasks without auxiliary student models or policy regularizers, making it a strong candidate for offline reinforcement learning.

Keywords: Offline Reinforcement Learning, Flow Matching, Generative Policies, Actor–Critic, Value Guidance

1 Introduction

Generative policies—diffusion [1], flows [2], and normalizing flows [3]—have expanded the function classes available to reinforcement learning (RL) beyond unimodal Gaussians. Yet, practical use remains complicated: many methods couple actor and critic losses tightly [4], require auxiliary students distilled from slow samplers [5], or rely on long-horizon planners that are cumbersome for control [6]. In parallel, Q-learning [7] remains a strong backbone for value estimation but does not by itself provide a rich, multi-modal policy class.

In response to this challenge, we propose **Q-Guided Flow Q-Learning (QFQL)**: a simple, robust actor–critic method that trains the *actor* as a conditional vector field with flow matching using pure behavior cloning, and trains the *critic* with standard temporal difference (TD) learning. At inference, we integrate the flow field and add a *value guidance* term $\beta \nabla_a Q(s, a)$, analogous in spirit to classifier-free guidance [8] in generative modeling but operating in action space and driven by Q rather than a classifier. This clean separation: (i) removes the need for joint actor–critic objectives during training, (ii) preserves the behavioral prior learned from data, (iii) enables a tunable value bias at test time that can be annealed or scheduled.

Contributions: We make the following contributions: (1) A decoupled training pipeline for generative actors and TD critics; (2) a value-guided sampling rule for actions requiring only $\nabla_a Q$ at inference; (3) empirical evidence that guidance improves control quality without retraining the actor.

2 Preliminaries

We consider a Markov decision process (MDP) with state space $\mathcal{S}$, action space $\mathcal{A}$, and reward space $\mathcal{R}$. A stochastic policy is a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ denotes the set of probability

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

measures over actions. The action–value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$Q(s,a) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \,\middle|\, s_0 = s,\ a_0 = a\right], \quad \gamma \in [0,1), \tag{1}$$

which evaluates the expected discounted return of starting from (s,a) and following π thereafter.

We assume an offline dataset $\mathcal{D} = \{(s,a,r,s')\}$ collected by unknown behavior policies. Two complementary approaches are standard in this setting: (i) *behavior cloning* [9], which directly learns the behavior policy π from $\mathcal{D}$, and (ii) *Q-learning* [7], which learns Q-functions to approximate long-term returns and enable policy improvement.

In our framework, the actor is parameterized as a conditional flow-matching model $v_\theta(s, a^\tau, \tau)$, where $\tau \in [0,1]$ indexes the flow. Starting from a Gaussian noise $a^{(0)} \sim \mathcal{N}(0, I)$, integrating the vector field progressively transports the distribution toward $p_{\text{data}}(a|s)$ as $\tau \to 1$. The critic $Q_\phi(s,a)$ complements this by evaluating long-horizon value. Training both components provides a balance between short-term imitation of demonstrated behavior and long-term value estimation.

3 Related Works

We organize related work by how the critic Q interacts with the generative policy: (i) no Q (pure behavior cloning), (ii) Q at training time, (iii) Q at inference time, and (iv) connections to classical policy gradients.

3.1 Generative Policies without Q

Several works use diffusion or flow-based models purely for behavior cloning. Diffusion policies [10] and flow-based policies [11] improve multimodality and calibration compared to Gaussian actors [12]. These methods demonstrate the strength of iterative samplers but lack mechanisms for value improvement. They provide the foundation upon which value-guided extensions are built.

3.2 Critic Used at Training Time

A large class of methods integrates Q into the training objective of the policy, which includes Diffusion-QL [13], DIAR [14], DreamFuser [15] and QVPO [16]. These methods entangle the actor and critic during optimization, requiring joint tuning and sometimes auxiliary losses. Furthermore, the sampling of actions from diffusion and flow models during training introduces backpropagation through time (BPTT), making it harder and more costly to train the model [5].

3.3 Critic Used at Inference Time

Inference-time guidance is inspired by classifier-free guidance in generative models [8]. Recent work shows that diffusion guidance can be interpreted as controllable policy improvement [17]. Flow Q-Learning (FQL) [5] comes closest to our setting: it trains a flow-matching actor and TD critic separately, but then distills the slow iterative sampler into a one-step policy for deployment. In contrast, QFQL preserves the iterative sampler and introduces a direct inference-time correction $a \leftarrow a + \Delta t(v_\theta + \beta \nabla_a Q)$, making the value trade-off tunable at test time without distillation.

Most prior methods use the critic either at training time or require student distillation for fast inference. QFQL occupies a distinct point: the actor and critic are trained entirely separately, and Q influences only inference through a tunable guidance term. This reduces training instability, avoids distillation, and exposes controllable trade-offs at **test-time**.

a QFQL Training

Require: Dataset $\mathcal{D}$; actor v_θ; critic Q_ϕ; target $\bar{Q}_{\bar{\phi}}$; γ, τ

while not converged **do**
  (Actor/FM) Sample $(s, a) \sim \mathcal{D}$; $t \sim \mathcal{U}(0,1)$; $\epsilon \sim \mathcal{N}(0, I)$
  $a_t \leftarrow (1-t)\epsilon + ta$; $u^\star \leftarrow a - \epsilon$
  $\theta \leftarrow \mathrm{Opt}_\theta\big(\theta, \nabla_\theta \|v_\theta(s, a_t, t) - u^\star\|^2\big)$
  (Critic/TD) Sample $(s, a, r, s') \sim \mathcal{D}$; $\tilde{a}' \leftarrow \mathrm{ApproxDenoisedAction}(s'; \theta)$
  $y \leftarrow r + \gamma\, \bar{Q}_{\bar{\phi}}(s', \tilde{a}')$
  $\phi \leftarrow \mathrm{Opt}_\phi\big(\phi, \nabla_\phi (Q_\phi(s, a) - y)^2\big)$
  $\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}$
end while

b QFQL Inference (Guided Flow)

Require: State s; actor v_θ; critic Q_ϕ; guidance β; steps K; schedule $\{(t_k, \Delta t_k)\}_{k=0}^{K-1}$

Initialize $a^{(0)} \sim \mathcal{N}(0, I)$
for $k = 0$ **to** $K-1$ **do**
  $g \leftarrow v_\theta(s, a^{(k)}, t_k)$;
  $\nabla Q \leftarrow \nabla_a Q_\phi(s, a^{(k)})$;
  $a^{(k+1)} \leftarrow a^{(k)} + \Delta t_k \big(g + \beta\, \nabla Q\big)$
end for
return $a^{(K)}$

(Optional): clip $\|\nabla Q\|$, anneal β.

Figure 1: Q-Guided Flow Q-Learning (QFQL): **training** (left) and **inference** (right).

4 Q-Guided Flow Q-Learning

In this section, we present **Q-Guided Flow Q-Learning**, which consists of actor training with behavior cloning and critic training with 1-step temporal-difference learning. The full algorithm is shown in Algorithm 1a,1b.

Actor via Conditional Flow Matching (Behavior Cloning). For $(s, a) \sim \mathcal{D}$, $t \sim \mathcal{U}(0,1)$, $\epsilon \sim \mathcal{N}(0, I)$, define a linear interpolation

$$a_t \triangleq (1-t)\,\epsilon + t\,a, \qquad u^\star(s, a_t, t) \triangleq a - \epsilon. \tag{2}$$

The actor minimizes

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}\big[\|v_\theta(s, a_t, t) - u^\star(s, a_t, t)\|_2^2\big], \tag{3}$$

purely from behavior data (no Q terms).

Critic via TD Learning. A target critic $\bar{Q}_{\bar{\phi}}$ provides bootstrapping. For $(s, a, r, s') \sim \mathcal{D}$, we use the standard TD-learning with recursive relationship as the following:

$$y \leftarrow r + \gamma\, \bar{Q}_{\bar{\phi}}(s', \tilde{a}'), \qquad \mathcal{L}_Q(\phi) = \mathbb{E}\big[(Q_\phi(s, a) - y)^2\big], \quad \bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}. \tag{4}$$

Inference with Value Guidance. Given s, initialize $a^{(0)} \sim \mathcal{N}(0, I)$ and integrate

$$a^{(k+1)} \leftarrow a^{(k)} + \Delta t_k \Big(v_\theta(s, a^{(k)}, t_k) + \beta\, \nabla_a Q_\phi(s, a^{(k)})\Big), \tag{5}$$

with a schedule $0 = t_0 < \cdots < t_K = 1$ and $\Delta t_k \triangleq t_{k+1} - t_k$. The scalar $\beta \geq 0$ trades off behavior adherence ($\beta \approx 0$) and value seeking (larger β). In practice, we can anneal β or clip $\|\nabla_a Q\|$ for stability.

5 Experimental Results

We use OGBench [18] with setups similar to Flow Q-Learning. Overall, we find that QFQL performs strongly across 5 OGBench tasks and 3 D4RL antmaze tasks without explicit actor-critic training as shown in Table 1. While reducing the number of parameters in comparison with FQL due to removal of one-step actor, QFQL maintains strong performance in several tasks including antsoccer and antmaze.

Table 1: **Performance comparison on selected environments.** For OGBench, We compare performances on default task (denoted by `(*)`) only. Results show mean ± standard deviation over multiple seeds for the first 5 OGBench single-task environments (marked with *) and 3 D4RL antmaze environments. We denote values at or above 95% of the best performance in bold, following OGBench [18].

	Gaussian	Diffusion	Flow Policies				
Task	BC	CAC	FAWAC	FBRAC	IFQL	FQL	QFQL
antmaze-large-navigate-singletask-task1-v0 (*)	0 ±0	42 ±7	1 ±1	70 ±20	24 ±17	80 ±8	**84**
antsoccer-arena-navigate-singletask-task4-v0 (*)	1 ±0	0 ±0	12 ±3	24 ±4	16 ±9	39 ±6	**44**
cube-double-play-singletask-task2-v0 (*)	0 ±0	2 ±2	2 ±1	22 ±12	9 ±5	**36** ±6	8
scene-play-singletask-task2-v0 (*)	1 ±1	50 ±40	18 ±8	46 ±10	0 ±0	**76** ±9	62
puzzle-3x3-play-singletask-task4-v0 (*)	1 ±1	0 ±0	1 ±1	2 ±2	0 ±0	**16** ±5	0
antmaze-umaze-diverse-v2	47	66 ±11	55 ±7	82 ±9	62 ±12	89 ±5	**94**
antmaze-medium-diverse-v2	1	0 ±1	44 ±15	**77** ±6	60 ±25	71 ±13	50
antmaze-large-diverse-v2	0	0 ±0	16 ±10	20 ±17	64 ±8	**83** ±4	52

Table 2: **Task-specific hyperparameters for offline RL.** We individually tune the hyperparamter β for each task on QFQL. The hyperparameters for other algorithms are taken from the original FQL paper.

	CAC	FAWAC	FBRAC	IFQL	FQL	QFQL
Task	η	α	α	N	α	β
antmaze-large-navigate-singletask-task1-v0 (*)	1	3	3	32	10	1
antsoccer-arena-navigate-singletask-task4-v0 (*)	1	10	30	64	10	1
cube-double-play-singletask-task2-v0 (*)	0.3	0.3	100	32	300	0.3
scene-play-singletask-task2-v0 (*)	0.3	0.3	100	32	300	0.3
puzzle-3x3-play-singletask-task4-v0 (*)	0.01	0.3	100	32	1000	–
antmaze-umaze-diverse-v2	0.01	3	10	32	10	0.3
antmaze-medium-diverse-v2	0.01	3	10	32	10	0.3
antmaze-large-diverse-v2	3.5	3	1	32	3	0.3

We find that β needs to be finetuned for each environment, similar to the hyperparameter α in FQL. We swept β with values $\{0.3, 1, 3, 10, 30, 100\}$ to ensure a balance in the weight of the guidance. This is presumably due to the fact that β controls the strength of the Q-guidance term $\beta \nabla_a Q$, similar to how the hyperparameter α controls the distillation (and hence the guidance) strength. We list all environment-dependent hyperparameters used for offline RL in Table 2.

We also discovered that QFQL fails to solve the puzzle-3x3 task. We provide a detailed analysis of this failure case and its implications for Q-guidance methods in Appendix A.

6 Conclusion

We presented *Q-Guided Flow Q-Learning (QFQL)*, a minimal actor–critic design that *decouples* training: the actor learns a conditional flow via behavior cloning; the critic learns via TD. At inference, a simple correction $\beta \nabla_a Q$ biases the flow integrator toward value without retraining or distillation. This separation reduces optimization entanglement, preserves behavioral priors, and allows controllable test-time trade-offs between realism and value. Results show that guidance consistently boosts returns over unguided flow actors with small computational overhead, suggesting a practical path to expressive, stable, and tunable policies in offline RL.

[0](*) indicates OGBench single-task environments representing default tasks from their respective task groups. The remaining environments are D4RL antmaze tasks.

References

[1] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[2] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[3] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[4] M. Alles, N. Chen, P. van der Smagt, and B. Cseke. Flowq: Energy-guided flow policies for offline reinforcement learning. *arXiv preprint arXiv:2505.14139*, 2025.

[5] S. Park, Q. Li, and S. Levine. Flow q-learning. *arXiv preprint arXiv:2502.02538*, 2025.

[6] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[7] R. S. Sutton, A. G. Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[8] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[9] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[10] 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*, page 02783649241273668, 2023.

[11] Q. Zhang, Z. Liu, H. Fan, G. Liu, B. Zeng, and S. Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14754–14762, 2025.

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[13] Z. Wang, J. J. Hunt, and M. Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning, 2023. URL https://arxiv.org/abs/2208.06193.

[14] J. Park, Y. Kim, S. Kim, B.-J. Lee, and S. Kim. Diar: Diffusion-model-guided implicit q-learning with adaptive revaluation, 2024. URL https://arxiv.org/abs/2410.11338.

[15] K. Luo, C. XIAO, Z. Huang, Z. Ling, Y. Fang, and H. Su. Dreamfuser: Value-guided diffusion policy for offline reinforcement learning, 2024. URL https://openreview.net/forum?id=9jmUwjZi7j.

[16] S. Ding, K. Hu, Z. Zhang, K. Ren, W. Zhang, J. Yu, J. Wang, and Y. Shi. Diffusion-based reinforcement learning via q-weighted variational policy optimization, 2024. URL https://arxiv.org/abs/2405.16173.

[17] K. Frans, S. Park, P. Abbeel, and S. Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

[18] S. Park, K. Frans, B. Eysenbach, and S. Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

A Analysis of Puzzle Task Failure

QFQL achieves zero performance on the puzzle-3x3 task across all experimental runs. This section presents a systematic analysis of training dynamics and evaluation behavior to identify the underlying causes of this failure.

A.1 Empirical Observations

Analysis of evaluation episodes and training dynamics reveals several distinctive patterns that characterize the failure mode. The policy exhibits alternating periods of minimal action output followed by control signals of extreme magnitude that result in NaN (Not-a-Number) values during flow integration. This instability manifests during training as gradient norms reaching maximum values of approximately 400, substantially exceeding the gradient norms observed in successful tasks which typically remain below 10. When the policy does produce valid actions, it frequently selects actions that leave the environment state unchanged (`button_states = prev_button_states`), resulting in no meaningful progress toward task completion. These observations suggest fundamental issues in both the policy's action generation mechanism and the underlying value estimation that guides it.

A.2 Causal Analysis

Two primary factors contribute to the observed failure modes. The first factor concerns reduced model capacity for combinatorial reasoning tasks. QFQL employs approximately two-thirds the parameters of FQL due to the elimination of the distilled one-step actor component. This architectural reduction presents particular challenges for combinatorial reasoning tasks such as the 3×3 puzzle environment, which contains $2^9 = 512$ possible state configurations, each requiring distinct value estimates and action mappings. Task success requires modeling multi-step action sequences where button press values depend on current configurations and future state transitions. Unlike continuous control domains, puzzle-solving demands discrete combinatorial reasoning patterns that may require substantial network capacity. The reduced parameter count may therefore be insufficient to represent the complex state-action mappings necessary for effective puzzle-solving behavior.

The second factor involves Q-function learning difficulties and the resulting degradation of guidance signals. The critic component demonstrates poor learning performance on the puzzle task, characterized by critic loss magnitudes several times larger than those observed in successful tasks, high variance in Q-value statistics (`q_min`, `q_mean`, `q_max`), and lack of convergence in Q-function estimates. When $Q_\phi(s, a)$ fails to provide accurate value estimates, the guidance term $\beta \nabla_a Q_\phi(s, a)$ contributes negatively to policy performance through multiple mechanisms. The gradient $\nabla_a Q$ provides directional signals uncorrelated with true value gradients, effectively misdirecting the policy during action generation. Large magnitude gradient signals contribute to the observed control signal explosions, while inaccurate Q-estimates compound through the guidance mechanism, further reducing policy effectiveness.

A.3 Factor Interaction and Methodological Implications

The identified factors interact to create a degradation cycle that explains the complete failure on puzzle tasks. Reduced network capacity constrains the critic's ability to learn accurate Q-functions for the combinatorial puzzle domain, leading to inaccurate Q-estimates that generate guidance gradients with high norms (approximately 400). These high-magnitude guidance signals disrupt the flow integration process, resulting in NaN values during action generation, which in turn provide inadequate training signals that further impair both actor and critic learning. This interaction pattern demonstrates that QFQL's decoupled training approach, while effective in continuous control domains, encounters fundamental limitations when both critic learning and network capacity are insufficient for the task complexity.

The analysis reveals several important constraints on inference-time Q-guidance approaches. Guidance-based methods may require increased rather than reduced model capacity for combinato-

rial reasoning tasks, contrary to the parameter reduction achieved by eliminating distillation components. Effective guidance depends critically on accurate Q-function learning, making the approach unsuitable for domains where value estimation is inherently difficult. Gradient norms exceeding 100 may serve as early indicators of guidance mechanism failure, suggesting the need for adaptive mechanisms that reduce guidance strength when instability is detected. Finally, Q-guidance demonstrates greater effectiveness in continuous control compared to discrete combinatorial domains, indicating that the approach's applicability may be fundamentally limited by the nature of the task domain rather than implementation details.