When Life Doesn't Give You Q-Values: Diagnosing Q-Exploitation in Test-Time Guidance of Flow-Matching Policies

Prajwal Avhad
Veermata Jijabai Technological Institute (VJTI)
Mumbai, India

***Abstract*—Test-time gradient guidance steers a frozen flow-matching policy toward higher-value actions using the gradient of a learned critic, without retraining the policy. We study this approach on a frozen diffusion-transformer policy (ABC-DiT) for a simulated bimanual bottle-placement task, training an in-sample critic on the policy's own visual features with a per-chunk ground-truth progress reward. We find that the binding constraint on guidance is the quality of the critic's action gradient, not the guidance mechanism. The critic separates successful from failed episodes well, but its action-gradient signal collapses to chance at mid- and late-trajectory horizons. We trace this failure to the action-outcome confound inherent in offline data, rather than to data imbalance, which we test and reject. In closed-loop evaluation, clean-action gradient guidance (QGF) increases the critic's predicted value while leaving real task performance at the noise floor, a failure mode we term Q-exploitation. The noisy-action variant (QFQL) instead yields a small but statistically significant improvement that a magnitude-matched random perturbation does not reproduce, establishing that the benefit is specific to the critic-gradient direction. Yet this improvement arises while the guidance moves the critic's predicted value the least: the benefit of test-time guidance here is decoupled from value ascent. We argue that an action-sensitive critic is a prerequisite for gradient guidance to translate predicted value into real performance, and offer Q-exploitation as a concrete diagnostic for when it will not.**

I. Introduction

Flow-matching and diffusion policies generate robot actions by integrating a learned velocity field from noise to a target action distribution [6, 20]. Once trained on demonstrations, the policy is frozen and deployed without parameter updates. Test-time gradient guidance offers a way to improve such a frozen policy: train a critic $Q(s, a)$ offline, then inject the critic's action gradient into the denoising loop during inference [7, 27].

Q-Guided Flow (QGF) [27] recently demonstrated this on OGBench manipulation tasks, outperforming prior test-time and training-time RL methods. QGF computes the critic gradient at a first-order Euler estimate of the denoised action, avoiding the cost and instability of backpropagating through the full denoising chain [13]. On OGBench [21], QGF benefits from large offline datasets (up to 1B transitions), dense reward, and low-dimensional state spaces with strong critic action gradients.

Whether the approach transfers to a setting closer to current robot learning practice is open. Visual policies trained on modest demonstration data produce rollout distributions that may not provide enough action diversity for a critic to learn informative action gradients. The critic must also operate on the policy's own visual conditioning features, introducing a representation bottleneck absent from state-based benchmarks. This paper studies when gradient-based steering of visual flow policies is valid, and proposes action-gradient quality and value-ascent decoupling as diagnostics for diffusion-policy guidance.

We study test-time Q-gradient guidance applied to ABC-DiT [2], a frozen 2B-parameter flow-matching diffusion-transformer policy for bimanual manipulation. The task is placing six plastic bottles into a bin in the ABC-Sim MuJoCo environment [25]. The policy takes three 224×224 camera images and 14D proprioception as input and outputs 30-step action chunks of absolute joint-position targets. We collect rollouts from the frozen policy in simulation, train a per-chunk critic using ground-truth bottle-count progress, and evaluate both QGF and QFQL in closed-loop.

We investigate three questions:

- **Q1.** Can a critic trained on a frozen visual policy's own features produce an informative action gradient?
- **Q2.** Does test-time gradient guidance translate the critic's predicted value into real task performance?
- **Q3.** Is any observed benefit specific to the critic-gradient direction, or generic perturbation?

We validate the critic pipeline before running guidance (**Q1**). The policy's pooled vision features are necessary for the state representation: proprioception alone is near chance at distinguishing good from bad outcomes early in execution. Ground-truth bottle-count reward produces a substantially stronger critic than Robometer [19], a 4B-parameter video-progress model, indicating that reward resolution is a bottleneck independent of the guidance mechanism. The resulting critic separates good from bad episodes, yet its action gradient is informative only early in execution and collapses to chance thereafter.

For **Q2**, clean-action QGF guidance reliably increases predicted Q without improving real task performance, a pattern we call **Q-exploitation**. The noisy-action variant QFQL produces a small but statistically significant improvement in partial progress. The method that moves predicted Q most

improves nothing; the method that improves real performance moves predicted Q least. We call this **value-ascent decoupling**.

For **Q3**, a magnitude-matched random perturbation does not reproduce the QFQL benefit, establishing that it is directional rather than perturbative.

These findings, scoped to this task and this critic, suggest that critic action-gradient quality is the binding constraint for test-time guidance of visual flow policies.

II. Related Work

a) Test-time RL for generative policies: Best-of-N sampling uses a critic to rank multiple policy samples [11, 18, 27]. It is effective but computationally expensive, requiring N full denoising rollouts per action. Gradient-based guidance modifies the denoising trajectory directly. QFQL [13] guides each step with $\nabla_{a_t} Q(s, a_t)$, the gradient at the noisy action; QSM [23] uses Q-score matching to similar effect. QGF [27] instead uses the gradient at a first-order approximation of the denoised action, achieving lower variance and better OGBench performance. Our work applies QGF to a visual manipulation policy and diagnoses a failure mode not visible on state-based benchmarks.

b) Offline RL for manipulation: Offline RL [17] learns policies from fixed datasets without environment interaction. IQL [15], CQL [16], TD3+BC [8], and BCQ [9] learn conservative value functions from offline data. These methods typically train the policy to maximize Q; we instead keep the policy frozen and apply Q gradients at test time, following the QGF paradigm.

c) Flow-matching robot policies: Diffusion Policy [6] introduced action diffusion for robot control, building on diffusion for planning [14] and offline RL [26]. π_0 [4] and $\pi_{0.5}$ [12] scale flow-matching VLA policies to diverse real-world tasks. ABC-DiT [2] is a diffusion-transformer policy trained on the ABC-130K dataset; we use its bottles-in-bin simulation checkpoint. LBM [3] studies architectural choices for large behavior models. Our work treats the policy as a frozen black box and focuses on what happens when a test-time critic is added.

d) Learned reward models for manipulation: Video-based progress and reward models provide dense supervision for long-horizon manipulation without hand-designed rewards. Stage-aware reward modeling [5] predicts within-stage progress, and the reward-as-progress-delta formulation it motivates computes per-chunk reward as the progress improvement over a policy chunk. Robometer [19] is a video-progress model in this family. We use such a progress signal to train our critic and additionally compare it against ground-truth task progress, finding that reward resolution—not only the guidance mechanism—bounds what test-time guidance can achieve.

III. Preliminaries

A. Flow Matching for Policy Learning

A flow-matching policy [20] parameterizes a time-dependent velocity field $v_\theta(x, t) : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ trained to transport noise $p_0 = \mathcal{N}(0, I)$ to the data distribution p_1 via the ODE $\mathrm{d}\hat{x}_t = v_\theta(\hat{x}_t, t)\,\mathrm{d}t$. Training minimizes the conditional flow matching loss over linear interpolants $x_t = (1 - t)\, x_0 + t\, x_1$:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t, x_0, x_1}\left[\|v_\theta(x_t, t) - (x_1 - x_0)\|_2^2\right]. \tag{1}$$

For policy learning, the velocity field is additionally conditioned on the state s, written $v_\theta(s, x_t, t)$, and trained on dataset actions as targets [6, 4]. At inference, actions are generated by integrating the ODE from $\hat{x}_0 \sim p_0$ to $\hat{x}_1$ over T discrete steps.

B. Test-Time Q-Gradient Guidance

We write a_t for the denoising variable x_t to emphasize that the generated quantity is an action chunk.

Given a frozen flow policy $\hat{\pi}$ and a learned critic $Q(s, a)$, the goal is to sample from the KL-regularized improved policy [1, 22] $\pi(a \mid s) \propto \hat{\pi}(a \mid s) \cdot \exp(Q(s, a)/\beta)$ at test time by modifying the denoising trajectory [27]. This amounts to adding a guidance term to the velocity at each denoising step t:

$$a_{t+\delta} = a_t + \delta\Big(v_\theta(s, a_t, t) + \tfrac{1}{\beta}\, g_t\Big), \tag{2}$$

where g_t is a gradient estimate of Q with respect to the action and $1/\beta$ controls guidance strength. The choice of g_t distinguishes the two variants we study:

a) QGF (clean-action gradient).: QGF [27] computes a first-order Euler estimate of the fully denoised action, $\hat{a}_1 = a_t + v_\theta(s, a_t, t) \cdot (1 - t)$, and evaluates the gradient there:

$$g_t^{\mathrm{QGF}} = \nabla_{\hat{a}_1} Q(s, \hat{a}_1). \tag{3}$$

This avoids querying the critic at out-of-distribution noisy actions and has lower variance than backpropagating through the denoising chain [27].

b) QFQL (noisy-action gradient).: The alternative [13] takes the critic gradient directly at the noisy intermediate action a_t:

$$g_t^{\mathrm{QFQL}} = \nabla_{a_t} Q(s, a_t). \tag{4}$$

Since the critic is trained only on denoised (clean) actions, this gradient is evaluated out of distribution, which can produce unreliable updates. QGF was introduced precisely to avoid this OOD issue, and it outperforms QFQL on OGBench tasks [27]. We evaluate both variants on a visual manipulation policy.

IV. Experimental Setup

A. Task and Policy

Figure 1 shows the full pipeline. We use the *throwing plastic bottles into a bin* task from ABC-Sim [2] (Figures 3 and 2), a MuJoCo-based simulation of the bimanual YAM robot station. The policy is a frozen ABC-DiT checkpoint (2B parameters) trained for 75k steps on the publicly released

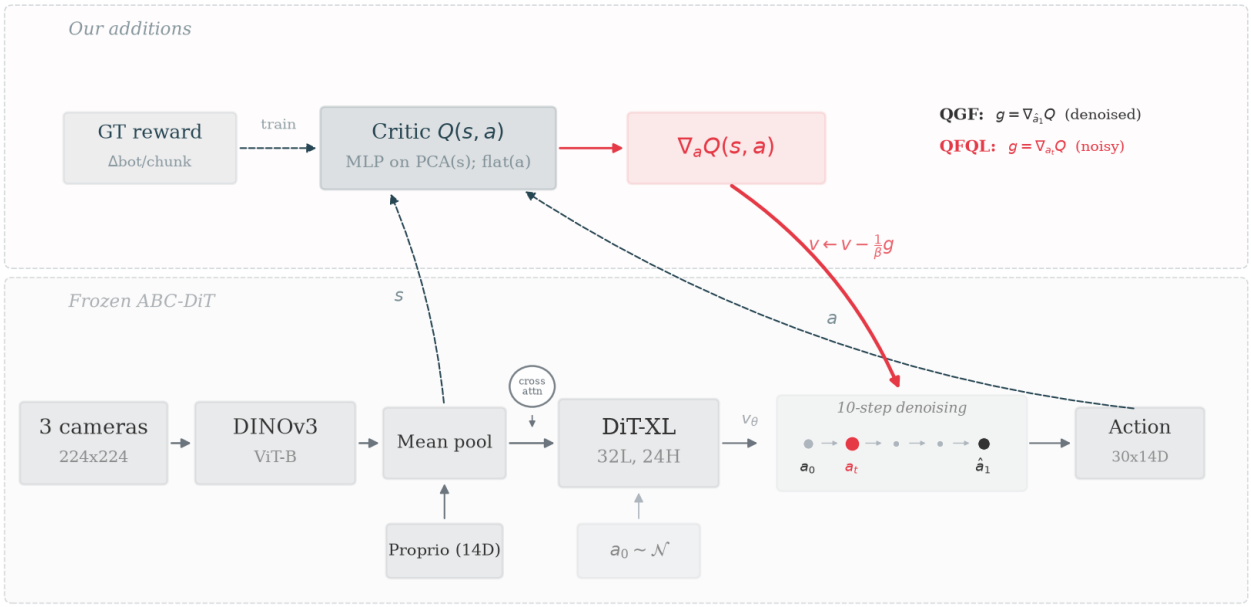

Fig. 1: System overview. Bottom (gray): the frozen ABC-DiT pipeline. Camera images are encoded by DINOv3 ViT-B, mean-pooled to produce state s, and cross-attention-conditioned into a DiT-XL that generates actions via a 10-step flow ODE. Top (red tint): our additions. A critic MLP trained with ground-truth bottle-count reward takes s and the action chunk a; its gradient $\nabla_a Q$ is injected into the denoising loop. QGF evaluates the gradient at $\hat{a}_1$; QFQL evaluates it at a_t.

(a) Failure (1/6) (b) Partial (4/6) (c) Success (6/6)

Fig. 2: Bottle-placement task in ABC-Sim. A bimanual YAM robot must place six plastic bottles into a bin. The frozen ABC-DiT policy achieves full success on 9% of episodes and places $\geq$5 bottles on 39%. Each panel shows three camera views (top, left, right) at representative outcomes.

ABC-130K dataset. It uses DINOv3 [24] cross-attention visual conditioning and outputs 30-step chunks of 14D absolute joint-position targets, normalized with z-score statistics. We run inference with 10 diffusion steps in eager (non-fast-inference) mode to expose intermediate denoising states.

a) Convention: The formulation in Section III-B (Eqs. 2–4) follows the standard flow direction $t: 0 \to 1$ with $v = x_1 - x_0$. ABC-DiT integrates in the reverse direction ($t: 1 \to 0$, $v =$ noise $-$ action, $\mathrm{d}t < 0$). We therefore use the convention-consistent clean-action estimate $\hat{a}_1 = x_t - t\,v$ and subtract the guidance term, $v \leftarrow v - \frac{1}{\beta}\,g$, so that a positive $1/\beta$ increases Q. We verified this sign empirically: guided actions attain higher critic value than unguided ones.

The task metric is `max_bottles_in_bin_so_far`, the peak bottle count reached during an episode. An episode runs for 120 action chunks (1800 environment steps). We report both strict success (6/6 bottles) and mean bottles placed.

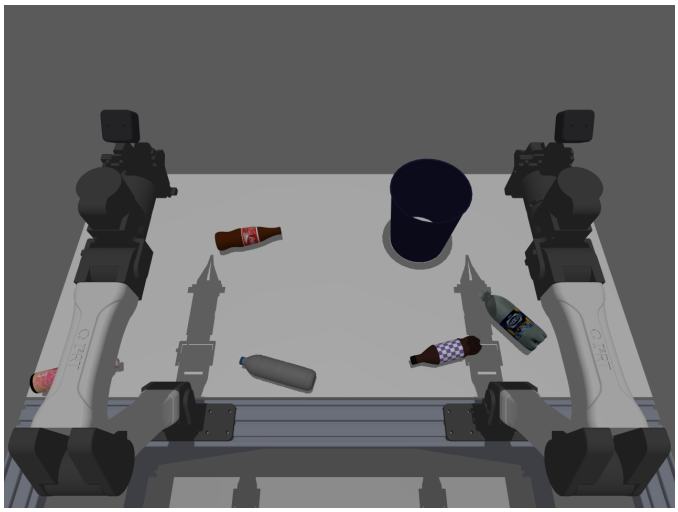

Fig. 3: Initial scene configuration. Three camera views (top, left, right) show the bimanual YAM robot station with six bottles scattered on the table and the target bin at center.

B. Data Collection

We collect 289 episodes from the frozen policy in the ABC-Sim MuJoCo environment with vanilla physics across distinct world seeds. Table I shows the outcome distribution (full histogram in Appendix H). The task occupies a partial-competence regime: the policy is neither too good nor too bad, providing natural variation for critic training.

At each action-chunk decision point, we record the policy's

TABLE I: Collection outcome distribution (289 episodes).

Bottles placed	0	1	2	3	4	5	6
Count	1	27	26	48	74	87	26
Percent	0.3	9.3	9.0	16.6	25.6	30.1	9.0

Mean: 3.84/6 (64%). Strict success (6/6): 9.0%. Good ($\geq$5): 39.1%.

pooled visual features (see below), the 30-step normalized action chunk, and the per-step ground-truth bottle count.

C. Critic

a) State representation: We use the frozen ABC-DiT vision encoder's pooled output as the state s. The encoder produces 36 DINOv3 tokens of dimension 1536 per image; we mean-pool across tokens and cameras to obtain $s \in \mathbb{R}^{1536}$. A linear probe confirms that pooled vision features separate good from bad episodes better than 14D robot state alone, especially early in execution (frame accuracy 0.62 vs. 0.54 at early horizon; 0.91 vs. 0.82 at episode level; Appendix D).

b) Action representation: The critic's action input is the same 30-step z-score-normalized action chunk that the policy outputs ($a \in \mathbb{R}^{30\times 14}$, flattened). This alignment is necessary; an earlier critic trained on variable-length raw action intervals was not compatible with guidance at policy-chunk granularity (Appendix G).

c) Architecture and training: The critic $Q(s, a)$ is a 3-layer MLP on $[\mathrm{PCA}_{128}(s)\,;\,\mathrm{flat}(a)] \in \mathbb{R}^{548}$. The per-chunk reward is the normalized bottle-count change, $r_k = \texttt{bottles}_{k+1}/6 - \texttt{bottles}_k/6$, following the progress-delta formulation used in stage-aware reward modeling [5]. The critic target is the discounted return-to-go advantage:

$$G_k = \textstyle\sum_{j\geq k} \gamma^{j-k} r_j, \qquad A_k = G_k - b_k, \tag{5}$$

where $\gamma = 0.99$ and b_k is a fitted per-chunk-position baseline that removes the time trend (so that $\nabla_a Q$ is not dominated by the clock). Training uses an episode-level held-out split. We do not bootstrap; the critic regresses A_k directly via MSE.

d) Anti-confound baselines: To verify that the critic learns from (s, a) jointly and not from clock or action-marginal correlations, we train matched baselines Q_{time} (chunk index + action), Q_{action} (action only), and Q_s (state only). The full critic $Q(s, a)$ outperforms these at the early horizon on both action-ranking and correlation (Table II, bottom).

e) Why ground-truth reward: We also trained critics using Robometer [19], a 4B-parameter video-progress model that scores rollout videos on a 0–1 scale. Robometer detects success, truncation, and reversal (Appendix B), but its per-chunk signal lacks resolution: it assigns similar scores to chunks that differ by one bottle. Table IV (Appendix C) compares critics trained on each reward source. GT bottle-count reward produces substantially higher mid-horizon action-ranking (0.70 vs. 0.57) and good/bad AUC (0.65 vs. 0.29). We use the GT-reward (rather than Robometer) critic for all guidance experiments. This choice means our results represent an upper bound on what current learned reward models can support: if guidance fails with ground-truth reward, it will not succeed with noisier video-based reward.

TABLE II: Guidance critic validation (held-out val set, n=1 415 chunks). This is the critic used for all guidance experiments: trained on policy-aligned 30-step normalized chunks at decision-frame granularity (13.5k rows, horizon $= f/1800$). Top: $Q(s, a)$ by horizon. Bottom: anti-confound baselines at the early horizon. Action-ranking: frequency $Q(s, a^+) > Q(s, a^-)$ for same-state pairs (chance 0.5). AUC: good ($\geq$5) vs. bad ($\leq$2) episodes. Probe: vision / robot-state frame accuracy.

	Act.-rank	Corr	AUC	Probe
$Q(s, a)$ by horizon				
Early (<0.3)	0.67	0.39	0.78	.62/.54
Mid (0.3–0.7)	0.53	0.35	0.83	.83/.70
Late ($\geq$0.7)	0.41	0.21	0.70	.91/.76
All	0.56	0.24	0.77	.75/.68
Anti-confound (early horizon)				
$Q(s, a)$	0.61	0.23		
Q_{time}	0.58	0.21		
Q_{action}	0.57	0.20		
Q_s (state only)	0.57	0.15		

Act.-rank = action-ranking (chance 0.5). AUC = good ($\geq$5) vs. bad ($\leq$2) episodes. Probe = vision / robot-state frame accuracy.

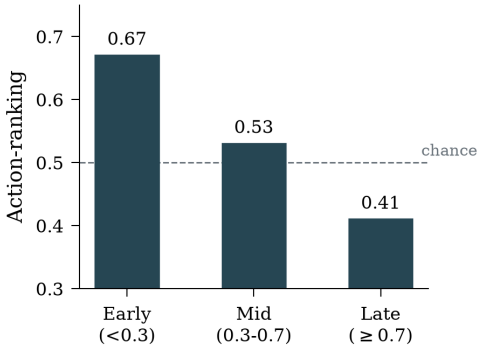

Fig. 4: State-conditioned action-ranking by trajectory horizon. The critic ranks actions above chance only early (< 0.3); mid and late horizons collapse to or below chance. Guidance via $\nabla_a Q$ can only help where ranking exceeds 0.5.

V. Results

A. Critic Validation (Q1)

Table II and Figure 4 answer **Q1**. The critic separates good from bad episodes well: mid-horizon good/bad AUC is 0.83. State-conditioned action-ranking, however, is 0.67 at the early horizon and drops to chance (0.53) at mid and below chance (0.41) at late (Figure 4). The critic knows which states are good but cannot reliably rank actions within a state at mid- and late-trajectory.

The state-probe column confirms that pooled vision features are necessary: at the early horizon, vision achieves 0.62 frame

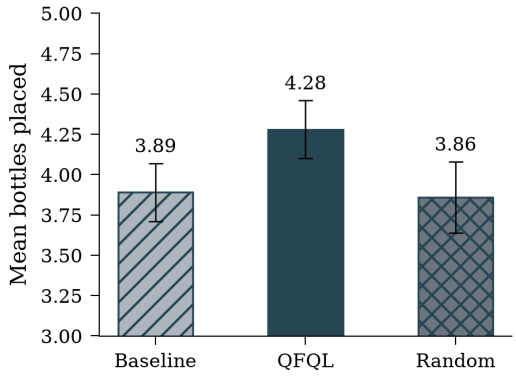

Fig. 5: Mean bottles placed (baseline / QFQL / random) with approximate standard errors. QFQL improves partial progress; random perturbation at matched $|g|/|v|$ does not (**Q3**).

accuracy vs. 0.54 for proprioception alone. The anti-confound baselines confirm that the full critic $Q(s, a)$ outperforms Q_{time}, Q_{action}, and Q_s at the early horizon. The critic is not exploiting clock correlations or action marginals.

This pattern is consistent with the offline action–outcome confound: the policy's action distribution at a given state is narrow (single-policy data), so the critic's signal for separating actions at a fixed state is weak. We tested and rejected the data-imbalance hypothesis: row-level training data is actually early-heavy (2809 rows at 0 bottles, 616 at 5 bottles), not mid-heavy (Appendix E).

For guidance to work via $\nabla_a Q$, the critic needs action-ranking above chance. This restricts the useful guidance window to early trajectory on this task.

B. Closed-Loop Guidance (Q2, Q3)

Table III reports all conditions. Figure 5 shows the headline comparison; Both QGF and QFQL guidance were applied in the early window ($t < 0.3$), the only horizon where the critic's action-ranking exceeds chance (0.67; Figure 4).

a) QFQL and the random control (Q2, Q3): QFQL produces a paired mean improvement of $+0.39$ bottles/world (nominally significant; 95% CI excludes zero, see caveat below). The per-world distribution is broad-based: 33 worlds improve, 27 are unchanged, 15 worsen (Appendix A), so the effect is not driven by a few outlier worlds. Strict task completion, however, is not improved: 12/75 successes vs. 10/75 for baseline, a difference indistinguishable from chance ($P(\geq 12 \mid p{=}0.133) = 0.30$). The benefit, such as it is, lies in partial credit only.

The random-perturbation control uses the same per-step $|g|/|v|$ as QFQL but replaces the gradient with a random unit vector; its paired Δbot is -0.04 (CI spans zero). Whatever QFQL contributes, it is specific to the critic-gradient direction rather than generic perturbation (**Q3**).

b) Caveats: The $n{=}75$ result is a single uncorrected paired t-test. We report it as nominally significant, not robustly established. A pre-registered replication at fixed n would strengthen the claim.

c) Q-exploitation (Q2): QGF reliably increases predicted Q: doubling the guidance weight from $1/\beta{=}16$ to $1/\beta{=}32$ roughly doubles ΔQ/step ($+0.024 \rightarrow +0.047$; Appendix F). Yet at the $n{=}16$ screen its paired bottle difference shows no positive signal and remains within the noise floor, while QFQL at the same screening budget did show one, which is why we extended QFQL rather than QGF. Predicted value goes up; real performance does not follow. We call this Q-exploitation: guidance climbs the critic's value surface without reaching states that are actually better.

C. Value-Ascent Decoupling

At the common $n{=}16$ screening budget, the ordering is already inverted: QGF moves predicted Q most ($+0.024$ to $+0.047$/step) yet shows no positive bottle effect, while QFQL moves Q least ($+0.010$) yet shows the positive signal that motivated its extension. The inversion is therefore not an artifact of unequal evaluation budgets; it appears at matched n.

In a well-calibrated critic, climbing predicted value should track real improvement; here the relationship is inverted even within the early window, the critic's best-case region. A ranking of 0.67 means the critic orders actions correctly only slightly more often than not. The clean-action gradient nonetheless steps precisely toward the critic's predicted optimum, so on the substantial fraction of states where that ordering is wrong, the precise step compounds the error: predicted Q rises while the action moves away from genuinely better ones. The noisy-action gradient, evaluated at intermediate a_t outside the critic's training distribution, provides a coarser directional nudge that is less tightly coupled to the critic's frequently mistaken optimum, and so degrades less and occasionally helps. When the critic's optimum is unreliable, precision hurts.

We did not guide indiscriminately: the offline action-ranking analysis (Section V-A) identified the early window as the only horizon with an above-chance action gradient, and we restricted guidance there. QGF's failure to improve is therefore not attributable to guiding where the critic is uninformative; it occurs in the critic's best-case region.

VI. Discussion

a) Why does QGF fail here when it works on OGBench?: The OGBench setting has three advantages absent here: (1) large offline datasets (100M–1B transitions from diverse policies) vs. single-policy data from 289 rollouts; (2) low-dimensional state-action spaces vs. 1536D visual state and 420D action chunks; (3) dense reward and stronger critics with action-ranking well above chance. The QGF gradient estimator is lower-variance than alternatives, but variance reduction does not help when the underlying gradient signal is at chance. We do not claim QGF is generally inferior to QFQL. We find that the clean-action advantage does not hold in this critic-limited regime.

TABLE III: Closed-loop guidance results. All conditions use $1/\beta$=16. Δbot is the paired per-world difference in max bottles vs. the seed-matched baseline. CI is the 95% confidence interval for the paired difference. ΔQ/step is the mean per-step change in predicted Q during denoising. $|g|/|v|$ is the gradient-to-velocity norm ratio. QGF conditions are from the n=16 diagnostic; QFQL and random from the pre-specified n=75 / n=50 endpoint.

Condition	n	Succ.	Mean bot.	Δbot	95% CI	t	ΔQ/step	$\lvert g\rvert/\lvert v\rvert$
Baseline	75	10/75	3.89	—	—	—	—	—
QFQL $1/\beta$=16	75	12/75	4.28	+0.39	[+0.09, +0.68]	2.59	+0.010	0.032
Random	50	3/50	3.86	−0.04	[−0.47, +0.39]	−0.19	0.000	0.032
Baseline (n=16)	16	2/16	3.50	—	—	—	—	—
QGF $1/\beta$=16	16	1/16	3.69	+0.19	[−0.52, +0.90]	0.56	+0.024	0.032
QGF $1/\beta$=32	16	2/16	3.81	+0.31	[−0.43, +1.06]	0.89	+0.047	0.065

b) What does the QFQL effect mean?: The improvement is small (+0.39 bottles on a 0–6 scale) and restricted to partial progress. It does not improve task completion. The nominally significant p-value warrants replication. We interpret the QFQL result as evidence that the critic-gradient direction carries some useful signal, not as evidence that QFQL provides practical policy improvement on this task.

c) Prerequisites for effective test-time guidance: Our results suggest two prerequisites, at minimum: (1) a critic whose state-conditioned action-ranking is above chance across the full trajectory (not just early), and (2) reward quality sufficient to train such a critic. We used ground-truth bottle counts; Robometer [19] (Appendix B) gave weaker critic diagnostics. Reward quality and critic action sensitivity are complementary bottlenecks, analogous to the reward-model overoptimization problem identified in language model alignment [10].

d) Limitations: This study uses a single task, a single policy, and a single critic architecture. The findings are diagnostic, not prescriptive. We do not claim that test-time gradient guidance cannot work for visual manipulation policies. We claim that it does not work here, and we identify where the failure sits.

VII. Conclusion

We applied test-time Q-gradient guidance to a frozen visual flow-matching policy for a simulated manipulation task. Clean-action QGF guidance exhibits Q-exploitation: it reliably increases the critic's predicted value while showing no corresponding gain in real task performance, even when confined to the early window where the critic's action gradient is most informative. The noisy-action variant QFQL produces a small improvement which is nominally significant and specific to the gradient direction, as a magnitude-matched random control does not reproduce it. The critic separates good from bad states but its action gradient is informative only early in execution.

On this task, critic action-gradient quality is the binding constraint, and reward resolution feeds it: a ground-truth progress signal yields a substantially stronger critic than a learned video-progress reward, indicating that reward quality is one upstream determinant of whether guidance can succeed. Value-ascent decoupling, the inversion between predicted-Q movement and real performance, serves as a diagnostic for this failure mode. Our results suggest that scaling test-time gradient guidance to visual manipulation policies may require critics with stronger action sensitivity, which in turn may require either diverse multi-policy data or critic architectures designed to separate state value from action value in narrow behavioral distributions.

References

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

[2] Arthur Allshire, Himanshu Gaurav Singh, Ritvik Singh, Adam Rashid, Hongsuk Choi, David McAllister, Justin Yu, Yiyuan Chen, Huang Huang, Pieter Abbeel, Xi Chen, Rocky Duan, Phillip Isola, Jitendra Malik, Fred Shentu, Guanya Shi, Philipp Wu, and Angjoo Kanazawa. Scalable behavior cloning with open data, training, and evaluation. *arXiv preprint*, 2026. URL https://abc.bot/.

[3] Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *Science Robotics*, 11(113):eaea6201, 2026.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[5] Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Yide Shentu, and Philipp Wu. Sarm: Stage-aware reward modeling for long horizon robot manipulation. *arXiv preprint arXiv:2509.25358*, 2025.

[6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via

action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[8] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[9] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[10] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

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

[12] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[13] Yejun Jang, Hong Chul Nam, Jeong Min Park, GIMIN BAE, and Hyun Kwon. Q-guided flow q-learning. In *CoRL 2025 Workshop RemembeRL*, 2025. URL https://openreview.net/forum?id=MFY9i3uR7S.

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

[15] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[16] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

[17] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[18] Qiyang Li, Zhiyuan Paul Zhou, and Sergey Levine. Reinforcement learning with action chunking. *Advances in Neural Information Processing Systems*, 38:55518–55553, 2026.

[19] Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S Huang, Luke Zettlemoyer, Dieter Fox, et al. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. *arXiv preprint arXiv:2603.02115*, 2026.

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

[21] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pages 94937–94982, 2025.

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

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

[24] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[25] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

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

[27] Zhiyuan Zhou, Andy Peng, Charles Xu, Qiyang Li, Jost Tobias Springenberg, Kevin Frans, and Sergey Levine. Test-time gradient guidance of flow policies in reinforcement learning. *arXiv preprint arXiv:2606.11087*, 2026.

APPENDIX A
PER-WORLD DELTA HISTOGRAM

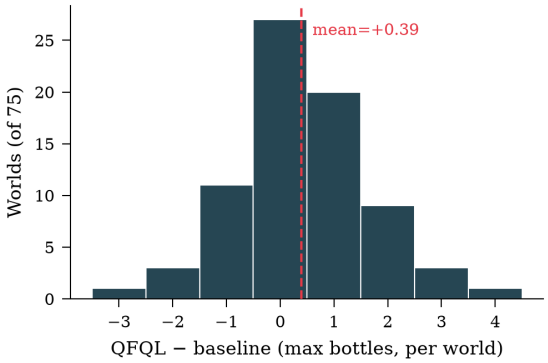

Fig. 6: Per-world Δbottles (QFQL minus baseline) for n=75 paired worlds. The distribution is broad: 33 positive, 27 zero, 15 negative. The effect is not driven by a small number of extreme worlds.

APPENDIX B
ROBOMETER VALIDATION

We trained critics using Robometer [19], a 4B-parameter video-progress model, as reward signal. Robometer scores video rollouts on a 0–1 progress scale. Initial experiments with the generic task description provided by ABC [2] produced near-uniform scores across good and bad rollouts; we obtained usable separation only after switching to a detailed task-completion prompt specifying the target outcome (all six bottles in the bin). We additionally ablated camera views (Figure 7) and validated sensitivity with controlled-failure rollouts (Figure 8).

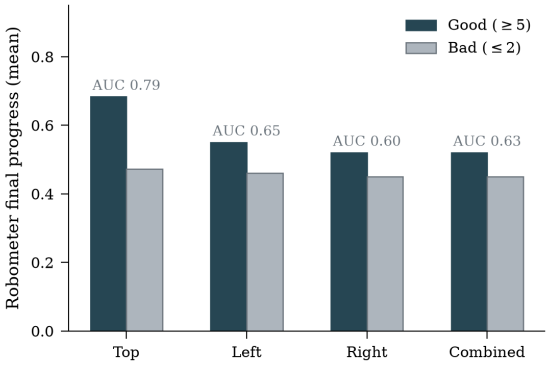

Fig. 7: Robometer view ablation. Good/bad separation by camera view. Top-view alone achieves the best separation (0.21 mean gap); combined three-view, left-only, and right-only are substantially weaker.

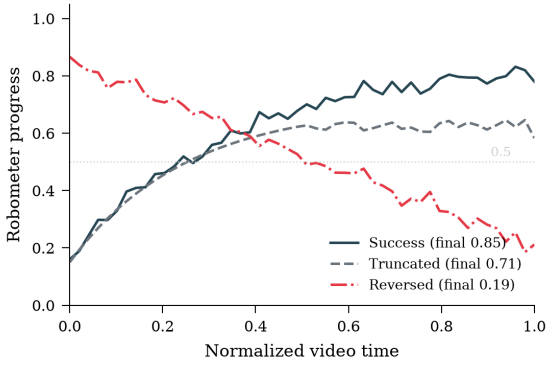

Fig. 8: Robometer controlled-failure diagnostic. Original successful rollout, truncated rollout (cut mid-task), and reversed rollout (played backwards). Robometer correctly detects progress, truncation, and reversal, confirming it is sensitive to task-relevant video content, not just visual complexity.

APPENDIX C
GT VS. ROBOMETER REWARD RESOLUTION

TABLE IV: Reward-source ablation: GT vs. Robometer reward ($Q(s,a)$ architecture, 15% held-out test set). Unlike Table II, these critics are trained on Robometer-grid chunks ($\leq$37-step raw actions, 3.5k rows, horizon $= k/30$) to isolate the reward-source effect at matched granularity. GT reward produces substantially higher mid/late action-ranking and AUC.

Reward	Horizon	Act.-rank	Corr	AUC
GT	Early	0.58	0.17	0.52
GT	Mid	0.70	0.25	0.65
GT	Late	0.48	0.17	0.66
GT	All	0.50	0.19	0.62
Rbm	Early	0.52	0.16	0.56
Rbm	Mid	0.57	0.06	0.29
Rbm	Late	0.53	−0.05	0.29
Rbm	All	0.62	0.07	0.36

GT un-blinds mid/late value (AUC 0.65–0.66 vs. 0.29). Robometer's video-level signal lacks per-chunk resolution.

APPENDIX D
STATE PROBE FULL BREAKDOWN

TABLE V: Good/bad classification probe accuracy by horizon and input type. Pooled vision features outperform 14D robot state at every horizon. The gap is largest early, where proprioception is near chance.

	Frame acc.		Episode acc.	
Horizon	Vision	Robot	Vision	Robot
All	0.75	0.68	0.91	0.82
Early	0.62	0.54	0.71	0.56
Mid	0.83	0.70	0.90	0.78
Late	0.91	0.76	0.93	0.82

APPENDIX E
DATA DENSITY AND RULED-OUT CONFOUNDS

TABLE VI: Training rows by current bottle count. The data is early-heavy (every episode passes through 0–2 bottles), not mid-heavy, ruling out data imbalance as the cause of mid-horizon critic weakness.

Bottles in bin	0	1	2	3	4	5
Rows	2809	3160	3055	2385	1454	616

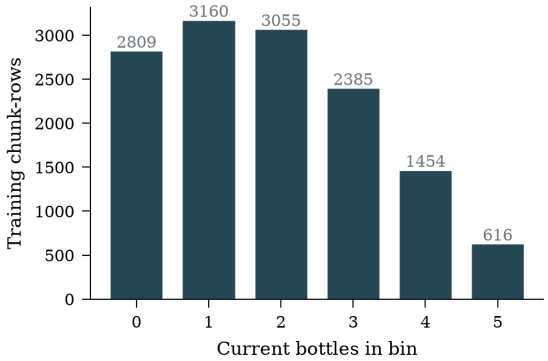

Fig. 9: Data density by current bottle count. The number of training chunks decreases monotonically with bottle count. Mid-horizon action-ranking weakness is not caused by data scarcity at those bottle counts.

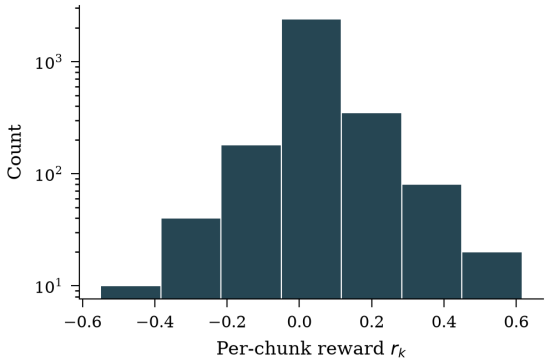

Fig. 10: Per-chunk reward distribution (pre-clipping). Most chunks have zero reward (no bottle-count change). The advantage target subtracts the episode mean to center this distribution.

APPENDIX F
GUIDANCE STRENGTH SWEEP

TABLE VII: Guidance strength sweep (n=16 paired worlds). These are full closed-loop rollouts (not per-chunk evaluations): each row is a complete 120-chunk episode with guidance applied in the early window. ΔQ/step scales with guidance weight; Δbot does not. This is Q-exploitation: the gradient moves predicted value without improving real task performance.

Condition	Succ.	Mean bot.	Δbot	t	ΔQ/step	$\lvert g\rvert/\lvert v\rvert$
Baseline	2/16	3.50	—	—	—	—
QGF $1/\beta$=16	1/16	3.69	+0.19	0.56	+0.024	0.032
QGF $1/\beta$=32	2/16	3.81	+0.31	0.89	+0.047	0.065
QFQL $1/\beta$=16	2/16	4.25	+0.75	2.42	+0.010	0.032

APPENDIX G
ACTION REPRESENTATION ALIGNMENT

An earlier critic (v1) was trained on variable-length raw action intervals rather than the policy's 30-step z-score-normalized chunks. This critic achieved one-step delta prediction above chance but was not compatible with guidance at policy-chunk granularity: the gradient $\nabla_a Q$ operated on a different action space than the denoising loop. Switching to the policy-aligned 30-step chunk representation resolved this mismatch and is a prerequisite for guidance.

APPENDIX H
COLLECTION OUTCOMES

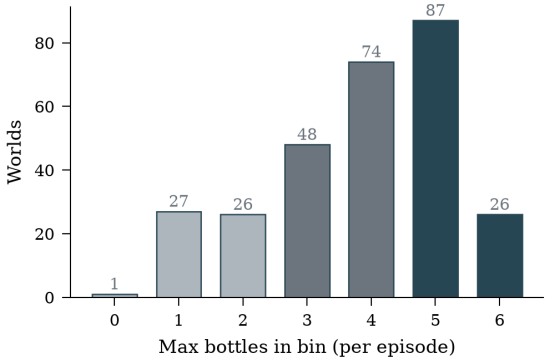

Fig. 11: Per-world max-bottles histogram for the 289-episode collection. Mean 3.84/6. Strict success (6/6): 26 worlds (9.0%). The task occupies a partial-competence regime.

(a) Failure (1/6) (b) Partial (4/6) (c) Success (6/6)

Fig. 12: Representative rollout outcomes from the frozen ABC-DiT policy. Each panel shows three camera views (top, left, right).