DATA-ANCHORED DYNAMIC ROUTING FOR ONE-STEP OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

One-step actors keep inference cheap in offline reinforcement learning, but still have to improve under a critic without moving too far from the data. In teacher–student methods such as FQL, the same student output is asked to do both jobs: move toward higher Q and stay near its paired teacher endpoint. When these directions disagree, the update is a compromise on that sample, even if another output could represent the target. We propose DROL, which replaces this teacher–student pairing with dynamic routing and trains the actor directly from offline data. For each state–action pair, the actor samples K candidates and updates only the nearest one with behavior cloning and critic guidance. As the actor changes, another candidate can take over the reconstruction target, allowing different latent regions to specialize without fixing their assignments in advance. Multiple candidates are needed only during training; inference uses a single actor pass. Our local analysis explains this assignment flexibility, and diagnostics track it in trained policies. On OGBench and D4RL, DROL is competitive with one-step baselines, with multi-candidate training improving performance across manipulation and puzzle task families.

1 INTRODUCTION

Offline Reinforcement Learning (RL) requires improving a policy while staying close to actions supported by a fixed dataset (Levine et al., 2020). This becomes harder on multimodal benchmarks such as OGBench (Park et al., 2025a), where a state can admit several distinct yet reasonable actions. Generative actors are attractive in this setting because they can represent these alternatives within a single policy (Sohl-Dickstein et al., 2015; Lipman et al., 2023; Wang et al., 2023; Hansen-Estruch et al., 2023; Park et al., 2025b). Their expressiveness, however, often comes with an iterative sampler that is costly to use during training or deployment.

A one-step actor avoids this sampling cost at inference. One way to train it, exemplified by FQL (Park et al., 2025b), is to learn a flow teacher from the data and distill it into a direct policy. For a latent draw z, the teacher produces an action $\tilde{a} = \mu_\psi(s, z)$, and the student output $f_\theta(s, z)$ is trained to improve Q while staying close to $\tilde{a}$. Within each update, the same output must therefore serve two roles: it is pushed toward higher Q and pulled back toward its paired teacher endpoint. When these directions disagree, the loss resolves the conflict by compromising on that sample.

For a multimodal actor, different samples can cover different parts of the data distribution. *Does a reconstruction target always need to be assigned to the same latent-conditioned output?* A candidate that moves away from a target may leave another candidate better placed to represent it. Pointwise distillation does not use this alternative: although the teacher can evolve, the pairing is still determined by the sampled latent. We instead let the actor's current outputs determine which candidate should receive the update, as illustrated in Figure 1.

DROL retains the one-step actor but replaces teacher–student pairing with data-driven routing. For each offline state–action pair, it samples K candidate actions, selects the one nearest to the data action, and updates only that winner with behavior cloning and Q-improvement. The assignment is recomputed at every update, so different data actions can supervise different latent regions, and another candidate can take over a target as learning proceeds. The winner still balances reconstruction against Q; what changes is which candidate must strike that balance. DROL thus trains directly from offline data, without learning or distilling an iterative teacher. At test time, it uses one random latent and one actor forward pass.

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

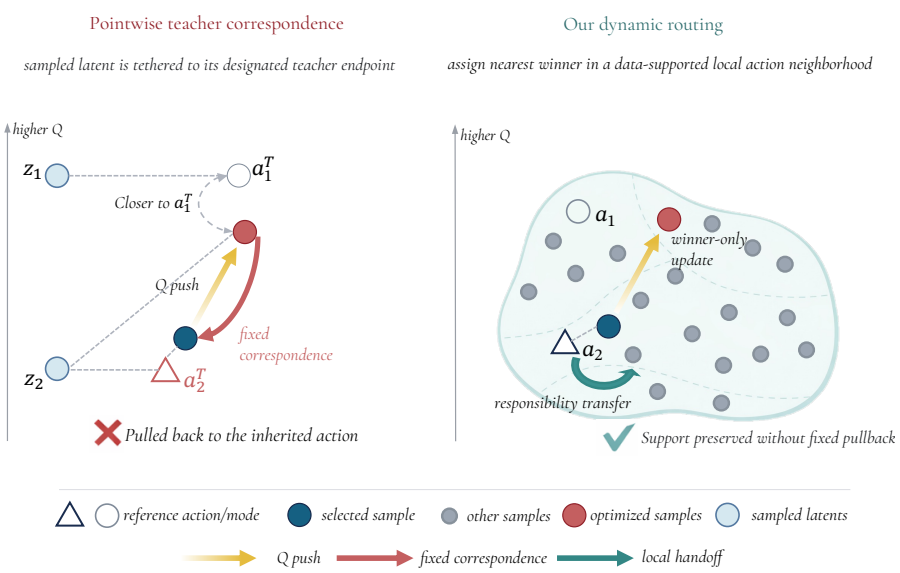


Figure 1: Let the reconstruction target change owners. Left: pointwise distillation pulls each student output toward its paired teacher endpoint while pushing it toward higher Q. Right: DROL assigns each data action to its nearest current candidate and applies both updates there. As the candidates move, another can take over the target.

Nearest-sample reconstruction is also used in IMLE and IMLE Policy (Li & Malik, 2019; Rana et al., 2025). Here, the data-to-candidate assignment selects not only what to reconstruct, but also where to apply critic guidance. Our analysis examines how candidates can specialize and how reconstruction targets can change owners; measurements on trained policies track these changes in practice. Experiments on OGBench and D4RL evaluate the resulting policy, with ablations showing that multiple candidates improve performance across manipulation and puzzle task families.

Our main contributions are:

- We propose DROL, a one-step offline actor that learns directly from data by routing each observed action to a candidate for behavior cloning and Q-improvement.
- We analyze how nearest-candidate routing encourages specialization and allows reconstruction targets to change owners, and examine these mechanisms in trained policies.
- We evaluate DROL on OGBench and D4RL, including candidate-budget ablations that demonstrate the benefits of multi-candidate training while retaining single-pass inference.

2 Preliminaries and Problem Setup

We consider a Markov decision process $(\mathcal{S}, \mathcal{A}, r, p, \gamma)$ and a fixed offline dataset $\mathcal{D} = \{(s, a, r, s')\}$. A critic Q_ϕ is trained by Bellman regression with a target critic and a sampled next action; the objective is specified in Equation (7). Our focus is the actor: it must improve predicted value while controlling deviation from observed behavior. DROL uses each dataset pair directly. Local neighborhoods are an analysis device, not an additional nearest-state retrieval step in the algorithm.

2.1 Pointwise-Correspondence One-Step Extraction

FQL (Park et al., 2025b) learns a behavior-cloned flow teacher μ_ψ and a one-step actor f_θ. Given shared latent $z \sim p_0$, the teacher provides $\tilde{a} = \mu_\psi(s, z)$. A generic extraction loss is

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{s,z}\left[-Q_\phi(s, f_\theta(s, z)) + \lambda\, \ell(f_\theta(s, z), \tilde{a})\right], \tag{1}$$

where ℓ is a reconstruction loss and λ is its weight. We use a separate symbol because the DROL objective below instead writes α as the coefficient of the Q term. The teacher is behavior-cloned, so

its endpoint is an approximate sample of supported behavior. The pointwise pairing assigns both reconstruction and improvement to one output within each update. Teacher endpoints can evolve over training; the correspondence is persistent as a pairing rule, not as a frozen action. DROL replaces this teacher-conditioned target with data-to-candidate matching.

Related approaches. Nearest-sample reconstruction is established in IMLE (Li & Malik, 2019) and IMLE Policy (Rana et al., 2025). DROL uses the selected candidate for both reconstruction and Q-improvement. ReFORM (Zhang et al., 2026) motivates a bounded latent prior through reflected flow constructions, whereas DROL trains a direct actor with a data-anchored objective. We do not claim the same support guarantees as a constrained flow construction. Further context appears in Section B.

3 Dynamic Routing for Offline RL

3.1 Actor as a Candidate Set

We start from a latent-conditioned actor $f_\theta(s, z)$. The local support around s is not observed directly, and exact support inclusion is intractable for a continuous generative actor. DROL therefore routes over a sampled candidate set drawn from the actor itself; the separate local-support object is used only in the conceptual discussion and visualization.

For each state s, we sample K latent codes and generate

$$\hat{a}^k = f_\theta(s, z_k), \qquad z_k \sim \mathrm{Unif}(B_R^{d_z}), \qquad k = 1, \dots, K, \tag{2}$$

where d_z is the latent dimension and $B_R^{d_z}$ is the d_z-dimensional Euclidean ball of radius R. Following ReFORM (Zhang et al., 2026), we adopt the dimension-aware choice $R = \sqrt{d_a}$ in our experiments, where d_a is the action dimension. With this choice, the overall latent scale stays on the same order as the action dimension (see Section C.5). These samples form the candidate set

$$A_\theta(s) = \{\hat{a}^1, \dots, \hat{a}^K\}. \tag{3}$$

A larger candidate set exposes more possible matches to the routing rule, providing opportunities for specialization and reassignment.

The choice $R = \sqrt{d_a}$ ties the latent scale to the action geometry rather than fixing a single radius across tasks, while keeping the prior isotropic. Compared with an unbounded Gaussian prior, $\mathrm{Unif}(B_R^{d_z})$ removes heavy-tail latent excursions. Compact latent support alone does not guarantee that generated actions lie in dataset support. At test time, the policy uses only a single latent sample,

$$\pi_\theta(s) = f_\theta(s, z), \qquad z \sim \mathrm{Unif}(B_R^{d_z}), \tag{4}$$

so the multiplicity K is purely a training-time device. After training, a single latent draw already samples from the reorganized actor, so execution stays within the plain one-step setting.

3.2 Top-1 Dynamic Routing

For each offline pair (s, a), DROL routes the dataset action to its current nearest candidate,

$$k^*(s, a) = \arg\min_{k \in [K]} \|\hat{a}^k - a\|_2^2. \tag{5}$$

Only the routed winner $\hat{a}^{k^*(s,a)}$ receives the actor update for that data point. Routing is recomputed from the current candidate geometry at every gradient step, and gradients are not propagated through the discrete $\arg\min$ itself.

The actor objective is

$$\mathcal{L}_{\mathrm{actor}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D},\, z_{1:K}} \Big[\|\hat{a}^{k^*(s,a)} - a\|_2^2 - \alpha Q_\phi\big(s, \hat{a}^{k^*(s,a)}\big) \Big]. \tag{6}$$

The first term is winner-only behavior cloning. It anchors the currently responsible candidate to the local data neighborhood that routing assigns to it. The second term performs critic-guided improvement on that same candidate. Both terms act on the same routed prototype, so the data

Algorithm 1 DROL

while not converged **do**
  Sample minibatch $\{(s_i, a_i, r_i, s'_i)\}_{i=1}^{B} \sim \mathcal{D}$
  Sample $z_{i,1:K} \sim \mathrm{Unif}(B_R^{d_z})$ and set $\hat{a}_i^k \leftarrow f_\theta(s_i, z_{i,k})$
  Route each data action to $k_i^* \leftarrow \arg\min_k \|\hat{a}_i^k - a_i\|_2^2$
  Update θ on $\frac{1}{B}\sum_i \left[\|\hat{a}_i^{k_i^*} - a_i\|_2^2 - \alpha Q_\phi(s_i, \hat{a}_i^{k_i^*})\right]$
  Sample $z'_i \sim \mathrm{Unif}(B_R^{d_z})$ and set $a'_i \leftarrow f_\theta(s'_i, z'_i)$
  Update ϕ on $\frac{1}{B}\sum_i \left[Q_\phi(s_i, a_i) - r_i - \gamma Q_{\bar{\phi}}(s'_i, a'_i)\right]^2$
  Soft-update target critic $\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}$

determines where critic guidance is applied. The reconstruction term anchors a candidate to the observed action without penalizing every other generated action.

Winner-only routing also has an optimization caveat: a latent region whose actions are never nearest to any minibatch action receives no direct per-sample gradient from the routing loss. In practice, the actor is a shared neural map over a compact latent ball, latents are resampled every update, and all reported evaluation uses the same one-sample policy rather than best-of-N filtering; these choices expose persistent unsupported regions empirically if they arise. This remains a limitation of hard top-1 routing and motivates softer or region-level variants.

The critic is trained with the standard Bellman loss

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\Big[\Big(Q_\phi(s,a) - \Big(r + \gamma \mathbb{E}_{a'\sim\pi_\theta(\cdot|s')}[Q_{\bar{\phi}}(s',a')]\Big)\Big)^2\Big]. \tag{7}$$

We estimate the inner expectation with a sampled one-step next action. Routing determines how support is regularized during actor learning, while execution and critic targets remain those of an ordinary one-sample policy.

3.3 Interpretation

Top-1 routing induces a state-conditional Euclidean Voronoi partition over the candidate set (Aurenhammer, 1991; Okabe et al., 2000). Given $A_\theta(s) = \{\hat{a}^1, \ldots, \hat{a}^K\}$, the cell of candidate k is

$$V_k(s) := \left\{a \in \mathbb{R}^{d_a} : \|\hat{a}^k - a\|_2^2 \le \|\hat{a}^j - a\|_2^2,\ \forall j \in [K]\right\}. \tag{8}$$

The cells $\{V_k(s)\}_{k=1}^K$ partition the action space up to measure-zero tie boundaries, and the routing rule returns the index of the cell containing the dataset action. Each dataset action therefore supervises only the candidate whose current cell contains it, and each candidate is responsible only for the portion of the local support that currently falls within its cell, namely $V_k(s) \cap \mathrm{supp}_{\mathrm{local}}(s)$ in the idealized local-support picture. If one candidate moves toward a better local action, another can inherit responsibility for the old neighborhood as soon as the nearest-neighbor partition changes. Iterative policies revisit several actions at inference time; DROL revisits several actions only during actor training. Execution uses ordinary one-sample generation, independently of how well candidate specialization has developed.

4 Local Analysis of Candidate Routing

We analyze three properties of multi-candidate training: specialization under reconstruction, changes in the owner of a joint BC/Q update, and the availability of alternative matches. The statements concern an idealized quantizer or local output-space geometry. They do not establish convergence of a shared neural actor, global support inclusion, or a performance ordering between algorithms. Proofs are deferred to Appendix C.

4.1 A Separated-Demand Toy Model Does Not Collapse

We first consider reconstruction without critic guidance in a separated one-dimensional distribution. This isolates the ability of nearest-candidate matching to represent multiple neighborhoods.

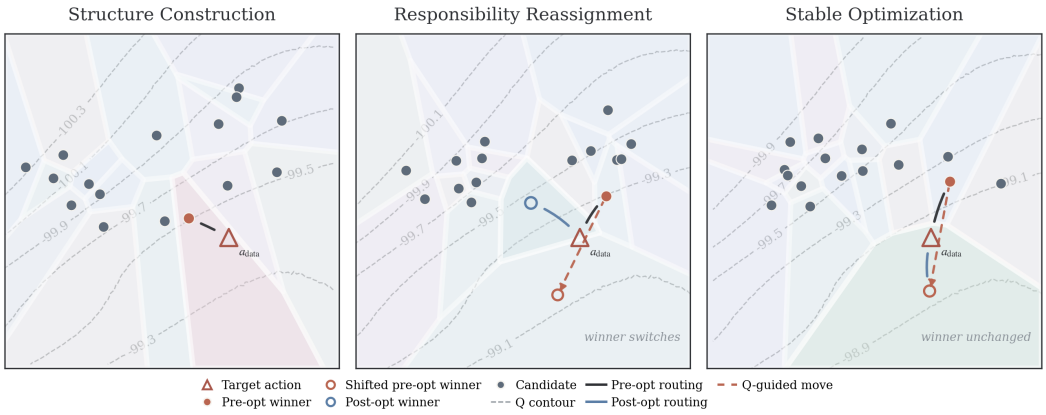


Figure 2: Mechanism visualization for DROL. The left panel shows structure construction, in which the candidate set spreads rather than collapses. The middle and right panels show two common local regimes after that structure has formed: responsibility reassignment and stable optimization. These two regimes are not meant to imply a fixed global time order. During training, different states can exhibit either behavior, and the same state can revisit both as the actor and critic continue to evolve.

Proposition 4.1 (Separated Neighborhoods Avoid Collapse)**.** *Let*

$$\rho = \frac{1}{M}\sum_{m=1}^{M} \mathrm{Unif}(I_m)$$

be a one-dimensional action distribution supported on disjoint intervals

$$I_m = [c_m - r, c_m + r], \qquad m = 1, \dots, M,$$

with center separations $c_{m+1} - c_m > 4r$ and $r > 0$. For $K = M$, consider the routed behavior-cloning objective

$$\mathcal{J}(\hat{a}_{1:K}) := \int \min_{k\in[K]} |\hat{a}_k - a|^2 \, \rho(\mathrm{d}a).$$

Then every global minimizer of $\mathcal{J}$ places exactly one prototype in each interval I_m. In particular, a collapsed configuration with two prototypes in one interval and none in another cannot be optimal.

This is a statement about global minimizers of an idealized reconstruction objective. It describes a specialization preference, but does not rule out poorly occupied regions or optimization plateaus in neural training. We examine candidate spread separately in Section 5.

4.2 Routing Replaces a Persistent Tether with a Transferable Anchor

Consider the gap between a pointwise-correspondence loss and a routed loss for a single supported target a at state s. Under pointwise distillation, a can be read as the designated teacher endpoint for a sampled latent-state pair. Under routing, it is the local target currently assigned to the winning candidate. For a single candidate x, the pointwise-correspondence objective is

$$\ell_{\mathrm{fix}}(x; a) = \|x - a\|_2^2 - \alpha Q_\phi(s, x).$$

For a routed candidate set $A = (x_1, \dots, x_K)$, define

$$\ell_{\mathrm{route}}(A; a) = \min_{k\in[K]} \|x_k - a\|_2^2 \; - \; \alpha Q_\phi\big(s, x_{k^*(a)}\big), \qquad k^*(a) = \arg\min_k \|x_k - a\|_2^2.$$

Proposition 4.2 (Persistent Tether versus Routed Anchor)**.** *For the pointwise-correspondence objective,*

$$\nabla_x \ell_{\mathrm{fix}}(x; a) = 2(x - a) - \alpha \nabla_x Q_\phi(s, x),$$

so the sample a continues to exert the same pullback term $2(x-a)$ regardless of what other candidates may exist. For the routed objective, if the winner $k^*(a)$ is unique, then

$$\nabla_{x_i} \ell_{\mathrm{route}}(A; a) = 0 \qquad \textit{for all } i \neq k^*(a),$$

and

$$\nabla_{x_{k^*(a)}} \ell_{\text{route}}(A; a) = 2(x_{k^*(a)} - a) - \alpha \nabla_x Q_\phi(s, x_{k^*(a)}).$$

Once another candidate becomes the nearest one to a, this behavior-cloning contribution transfers to the new winner.

For a fixed winner, the local BC/Q compromise is unchanged. The flexibility lies in which candidate receives it when the assignment changes; both terms in DROL then act on the new winner. These are direct output gradients. Shared parameters can change all outputs indirectly, and the full routed objective can be discontinuous at distance ties if the candidates have different Q-values. In FQL, the teacher itself can evolve during training; a persistent correspondence refers to the latent-conditioned pairing rule, not an endpoint frozen for all time.

Proposition 4.3 (Pointwise Correspondence Biases a Local Improvement)**.** *Let $U \subset \mathbb{R}^{d_a}$ be a closed convex local analysis neighborhood and $\alpha > 0$. Suppose $Q_\phi(s, \cdot)$ is differentiable and m-strongly concave on U, and $x_Q \in U$ is an interior maximizer over U. For a target $a \in U$, define*

$$g(x) = \|x - a\|_2^2 - \alpha Q_\phi(s, x), \qquad x \in U.$$

Then g has a unique minimizer x_{fix}. Moreover:

- *x_Q is stationary for g if and only if $a = x_Q$.*
- *The minimizer satisfies*

$$\|x_{\text{fix}} - x_Q\|_2 \le \frac{2}{2 + \alpha m} \|a - x_Q\|_2.$$

Strong concavity is a local sufficient condition, not a global property assumed of learned critics. The first claim establishes nonzero bias when $a \neq x_Q$; the second bounds its magnitude from above, not below. Both apply to a routed output with a fixed assignment. Together, Propositions 4.2 and 4.3 distinguish the compromise within an assignment from the possibility of changing that assignment. Whether such changes occur and improve performance is an empirical question; Section 5.3 measures their occurrence, and the candidate ablation measures the effect of the overall training design.

4.3 Larger K Exposes More Matching Opportunities

The final question is why a larger routing budget helps on some task families. For a fixed state s, suppose the local support is covered by modal neighborhoods $U_1(s), \ldots, U_{M_s}(s)$. Let

$$p_m(s) := \mathbb{P}_{z \sim \text{Unif}(B_R^{d_z})}\big(f_\theta(s, z) \in U_m(s)\big),$$

so that $p_m(s)$ is the probability that one sampled candidate lands in neighborhood m.

Proposition 4.4 (Coverage from Larger Candidate Sets)**.** *If $p_m(s) > 0$ for all $m = 1, \ldots, M_s$, then*

$$\mathbb{P}\Big(\forall m, \exists k \in [K], \hat{a}_k \in U_m(s)\Big) \ge 1 - \sum_{m=1}^{M_s} (1 - p_m(s))^K.$$

In particular, letting $p_{\min}(s) := \min_m p_m(s)$,

$$\mathbb{P}\Big(\forall m, \exists k, \hat{a}_k \in U_m(s)\Big) \ge 1 - M_s e^{-K p_{\min}(s)}.$$

This is not a coverage-creation statement. Increasing K does not invent new supported actions; it only makes it less likely that the current candidate set misses neighborhoods that are already reachable. That matters because responsibility transfer only works if another candidate is available nearby to take over. This is the geometric role of K in DROL.

5 Experiments

We evaluate one-step policy performance on OGBench (Park et al., 2025a) and D4RL (Fu et al., 2020), then examine candidate specialization, responsibility changes, and the role of update selectivity. The performance ablation and mechanism diagnostics answer different questions: the former measures the effect of candidate-based training, while the latter checks whether the local phenomena in Section 4 occur on benchmark-trained actors.

Table 1: **Offline RL results on OGBench and D4RL.** DROL(16) uses the fixed default $K = 16$ throughout. DROL* uses a benchmark-specific K-selection protocol; see Appendix A for the exact OGBench and D4RL settings. Bold entries are at least $0.95\times$ the best score in their row, and underlined entries are one-step methods that are at least $0.95\times$ the best score. See Table 4 for the full task-level version and Figure 4 for representative K-sweep trends.

	Gaussian Policies			Multi-step / Iterative Policies†					One-step Methods		
Task Category	BC	IQL	ReBRAC	FAWAC	FBRAC	IFQL	DeFlow	SORL	FQL	DROL(16)	DROL*
antmaze-large	11 ±1	53 ±3	81 ±5	6 ±1	60 ±6	28 ±5	81 ±3	**89** ±2	79 ±3	**90** ±2	**89** ±6
antmaze-giant	0 ±0	4 ±1	26 ±8	0 ±0	4 ±4	3 ±2	12 ±5	9 ±6	9 ±6	3 ±3	**45** ±24
humanoidmaze-medium	2 ±1	33 ±2	22 ±8	19 ±1	38 ±5	60 ±14	48 ±4	64 ±4	58 ±5	**65** ±9	**68** ±7
humanoidmaze-large	1 ±0	2 ±1	2 ±1	0 ±0	2 ±0	**11** ±2	5 ±2	5 ±2	4 ±2	7 ±4	**11** ±6
antsoccer-arena	1 ±0	8 ±2	0 ±0	12 ±0	16 ±1	33 ±6	**67** ±3	**69** ±2	60 ±2	62 ±2	65 ±7
cube-single	5 ±1	83 ±3	91 ±2	81 ±4	79 ±7	79 ±2	**96** ±2	**97** ±1	**96** ±1	**97** ±2	**98** ±2
cube-double	2 ±1	7 ±1	12 ±1	5 ±2	15 ±3	14 ±3	**40** ±5	25 ±3	29 ±2	20 ±3	26 ±7
scene	5 ±1	28 ±1	41 ±3	30 ±3	45 ±5	30 ±3	51 ±3	**57** ±2	**56** ±2	54 ±3	**58** ±3
puzzle-3x3	2 ±0	9 ±1	21 ±1	6 ±2	14 ±4	19 ±1	**43** ±4	–	30 ±1	**41** ±9	**41** ±9
puzzle-4x4	0 ±0	7 ±1	14 ±1	1 ±0	13 ±1	**25** ±5	11 ±2	–	17 ±2	16 ±2	**25** ±6
D4RL AntMaze	17	57	78	44 ±3	64 ±7	65 ±7	**83** ±2	–	**84** ±3	**81** ±6	**81** ±6
D4RL Adroit	48	53	**59**	48 ±1	50 ±2	52 ±1	52 ±1	–	52 ±1	51 ±3	52 ±3

[1] DROL* uses the benchmark-specific K-selection protocol described in Appendix A; all other training settings are unchanged from DROL(16). [2] The group headed by † denotes multi-step or iterative policies.

5.1 Main Results on OGBench and D4RL

Table 1 compares policies using a single latent draw at deployment. The fixed-budget DROL(16) is competitive with FQL across OGBench, while family-specific candidate selection further improves several groups. For OGBench, we first fix $K = 16$ and select α with the same search scope and budget as FQL, reusing its selected settings in most families. We then freeze α and, for DROL* only, select K on one designated task per family. The selected K is reused on the other four tasks; the extra search is part of the DROL* budget. The corresponding D4RL protocol is specified in Appendix A.

The clearest multi-candidate effects occur in manipulation and puzzle tasks, as examined in the candidate-budget study below. D4RL provides complementary evidence that the same actor design remains competitive on Adroit and AntMaze. The runtime comparison in Figure 3 shows the deployment benefit: DROL uses one direct actor pass, with inference latency close to FQL and below the measured iterative baselines.

5.2 Candidate Specialization and Assignment Geometry

Figure 2 illustrates candidate spread, responsibility reassignment, and locally unchanged assignments using a dense surrogate at one representative state. It is a mechanism illustration; these regimes can recur in any order during training. We separately measure geometry on benchmark-trained actors.

In Figure 3, pairwise candidate spread and centroid divergence increase with K, while routed reconstruction loss decreases. These observations are consistent with the specialization preference of Proposition 4.1. They do not establish recovery of every behavior mode. On final checkpoints from cube-double task 2, Scene task 2, and puzzle-4x4 task 4, the paired-action RMSE of the nearest candidate decreases from 0.062 to 0.052, 0.064 to 0.055, and 0.058 to 0.047, respectively, between $K = 16$ and 64. Only 19.6–21.6% of probes have a second-best squared distance within 10% of the winner. These four-seed measurements use 4,096 fixed transitions per seed. A larger candidate bank also mechanically improves nearest-sample matching, so the distance trend alone cannot isolate a learned specialization effect. Full settings and single-draw distances are in Section E.

5.3 Benchmark Gradient Conflict and Responsibility Changes

Conflict in a pointwise update. We train released FQL on Scene task 2 and AntMaze-Large task 1 with two prespecified seeds and inspect four checkpoints from 250k to 1M. On 4,096 fixed transitions per seed, the Q-ascent direction and the direction toward the teacher endpoint disagree on 81.7–83.5% of Scene transitions and 69.5–71.9% of AntMaze transitions. The mini-batch Q-loss and distillation-

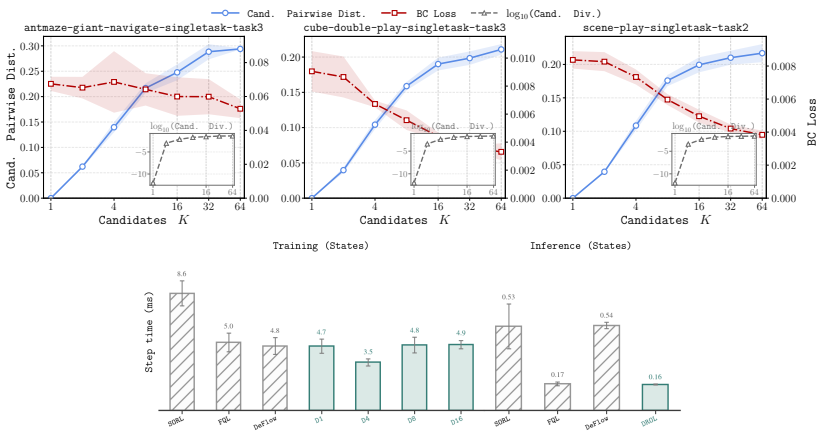


Figure 3: Candidate statistics and runtime. Top: normalized pairwise squared distance, centroid divergence (inset), and routed reconstruction loss on three representative environments. Bottom: state-input training and inference times on an A100. Training labels `D1`/`D4`/`D8`/`D16` denote $K \in \{1, 4, 8, 16\}$; inference uses one DROL bar because it always samples one action (0.16 ms, versus FQL's 0.17 ms).

loss gradients restricted to the one-step actor have negative cosine in all 16 fixed batches at each checkpoint. This verifies that the local compromise in Proposition 4.3 occurs in benchmark-trained FQL; it does not show that DROL reduces instantaneous conflict or that conflict alone determines return.

Changes of responsibility. For DROL($K = 16$) on Scene task 2, we track 4,096 fixed targets and one fixed latent bank at 50k-spaced checkpoints over 1M updates, using two seeds. After excluding initialization, 99.8% of targets change nearest owner at least once, and 76.1% experience a change that persists for two subsequent checkpoints. At the event level, however, only 11.0% of changes persist this long: reassignment is frequent and often temporary. These are checkpoint-scale changes for fixed probes, not literal training-time handoff counts, since training resamples latents. Both diagnostics use no policy evaluation and do not measure a causal effect on performance; detailed protocols appear in Section E.

5.4 Controlling Update Selectivity with K

The candidate budget controls where the joint BC/Q update is applied, not just how many actions are sampled. More candidates give each data action more alternatives for assigning its update; fewer candidates apply guidance less selectively over sampled latents. Unlike the BC/Q weight, K changes which output receives both terms, although the resulting gradients also depend on that choice. Figure 4 studies this control at fixed one-sample inference: several manipulation and puzzle tasks favor $K > 1$, whereas some navigation tasks prefer smaller budgets. To test whether the routing gains extend beyond individual sweep tasks, we evaluate $K = 1$ and $K = 16$ on all five tasks of cube-double, Scene, and puzzle-4x4. With four matched seeds, fixed family settings, final-1M evaluation, and 50 episodes, $K = 16$ improves the family means by 17.0, 21.2, and 11.6 success points, respectively. All four paired seed differences in each family favor $K = 16$. The full means and uncertainty estimates are reported in Table 5.

Why can fewer candidates help? At $K = 1$, every sampled output receives BC and Q guidance without a nearest-candidate filter; this remains a regularized objective. Such broader guidance can be preferable to concentrating updates on good data matches. For example, the full AntMaze-Giant family favors $K = 1$ over $K = 16$ (45.1 versus 2.7). The result is consistent with the tradeoff between broader guidance and selective matching, but does not identify its cause. Nor is $K = 1$ uniformly stable: cube-double task 1 falls from 38.5% at 100k to 8.0% at 1M.

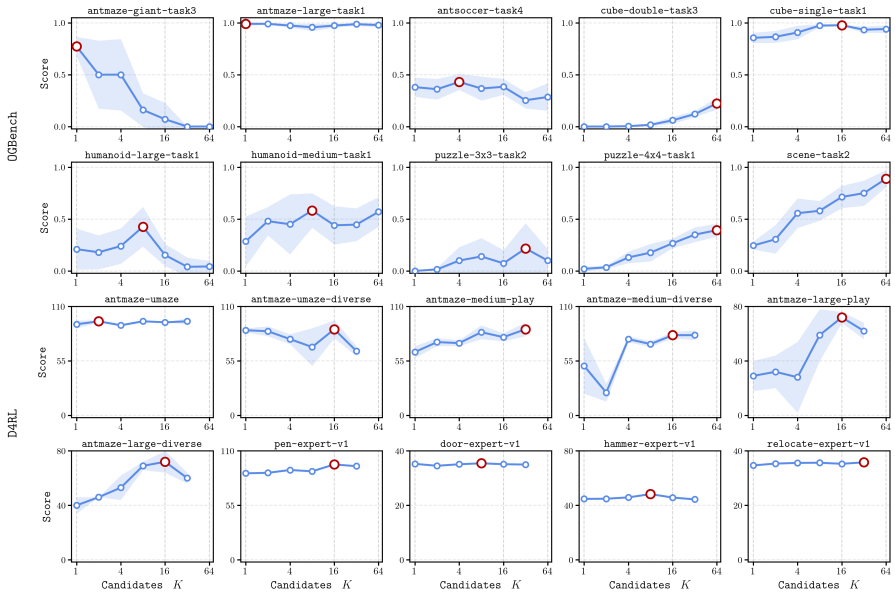


Figure 4: Varying the training-time candidate budget, with single-sample inference throughout. The top ten panels show success rates on one task per OGBench family; the remaining panels show D4RL normalized scores. Each panel sweeps $K \in \{1, 2, 4, 8, 16, 32, 64\}$ when available. The dashed line marks $K = 16$ and the red circle marks the best observed value within that panel. These diagnostic panels need not be the tasks used for family-level hyperparameter selection.

Using the control. These results suggest adapting the selectivity of the update, rather than simply increasing K. We hold α fixed when varying K and transfer one selected budget across tasks in a family. Selection uses evaluation rollouts, not a data-only criterion; its additional cost belongs to DROL* rather than the fixed-K reference.

6 Conclusion and Limitations

DROL trains a one-step offline policy using data-anchored routing over multiple candidate actions. The dataset selects which output receives the joint reconstruction and Q-improvement update, allowing candidate specialization and changes of responsibility while retaining single-sample inference. Local analysis explains these possibilities, and benchmark ablations show that multiple candidates can improve performance across complete manipulation and puzzle families.

The benefits are task dependent: $K = 1$ remains effective in some environments, while additional candidates offer selective updates and more matching opportunities. Our local theory does not establish neural actor-critic convergence or support guarantees for every generated action. Shared parameters can affect rarely winning latent regions without formal protection, and fixed-probe diagnostics do not establish causal effects on return. The choice of K, latent radius, and BC/Q scaling remain practical limitations.

Test-time scaling. DROL defaults to single-sample inference. Training-time candidate coverage does not ensure improved decisions from additional test-time samples; Q-based reranking also depends on critic accuracy. Reliable scaling with inference budget remains open.

Routing geometry. Euclidean matching is isotropic and does not explicitly account for behavior-density geometry. FiDec (Cheng et al., 2026) motivates Fisher-based geometry for local policy refinement; adapting such geometry to DROL's discrete routing, and establishing its benefits, remain open questions.

AI USE STATEMENT

Generative AI tools were used to discuss and refine the presentation of the method and its hypotheses, provide feedback on experimental design, assist with experiment-launch and diagnostic scripts, summarize and interpret experimental records, and edit the manuscript and its LaTeX formatting. The authors are responsible for verifying the analyses, mathematical claims, references, and final text.

REPRODUCIBILITY STATEMENT

The actor objective and training procedure are given in Section 3. Architecture, optimization, and evaluation settings are described in Appendix A; mathematical derivations are in Appendix C. The anonymous code supplement contains the agent, training entry point, OGBench/D4RL data-loading utilities, and example commands. The complete benchmark breakdown and the protocols, seed counts, and aggregation rules for the supplementary routing studies appear in Section E and the preceding task-level tables.

REFERENCES

Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991. doi: 10.1145/116873.116880.

Jongseong Chae, Jongeui Park, Yongjae Shin, Gyeongmin Kim, Seungyul Han, and Youngchul Sung. Flow actor-critic for offline reinforcement learning. In *International Conference on Learning Representations*, 2026.

Xiaoyuan Cheng, Haoyu Wang, Wenxuan Yuan, Ziyan Wang, Zonghao Chen, Li Zeng, and Zhuo Sun. Fisher Decorator: Refining flow policy via a local transport map. *arXiv preprint arXiv:2604.17919*, 2026.

Nicolas Espinosa-Dice, Yiyi Zhang, Yiding Chen, Bradley Guo, Owen Oertell, Gokul Swamy, Kiante Brantley, and Wen Sun. Scaling offline RL via efficient and expressive shortcut models. *arXiv preprint arXiv:2505.22866*, 2025.

Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pp. 158–168, 2022.

Justin Fu, Aviral Kumar, Ofir Nachum, G. Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *ArXiv*, abs/2004.07219, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2021.

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*, abs/2304.10573, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Aviral Kumar, Aurick Zhou, G. Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

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

Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. In *International Conference on Learning Representations*, 2019.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhancun Mu et al. DeFlow: Decoupling manifold modeling and value maximization for offline policy extraction. *arXiv preprint arXiv:2601.10471*, 2026.

Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2 edition, 2000. doi: 10.1002/9780470317013.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations (ICLR)*, 2025a.

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. *ArXiv*, abs/2502.02538, 2025b.

Krishan Rana, Robert Lee, David Pershouse, and Niko Suenderhauf. IMLE Policy: Fast and sample efficient visuomotor policy learning via implicit maximum likelihood estimation. In *Robotics: Science and Systems*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015.

Zeyuan Wang, Da Li, Yulin Chen, Ye Shi, Liang Bai, Tianyuan Yu, and Yanwei Fu. One-step generative policies with Q-learning: A reformulation of meanflow. *arXiv preprint arXiv:2511.13035*, 2025.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Shiji Zhang et al. ReFORM: Reflected flows for on-support offline rl via noise reflection. In *International Conference on Learning Representations*, 2026.

A Implementation Details

We use the same offline training and evaluation pipeline as the one-step FQL baseline. For each state s, DROL samples K latent noises uniformly from the radius-$\sqrt{d_a}$ ball, maps them through the one-step actor MLP, and obtains K candidate actions. The dataset action is routed to its nearest candidate under squared Euclidean distance, and both the BC term and the Q-improvement term are applied only to that winning candidate. All reported results use one-step inference. We do not use Q-based best-of-N action selection at deployment; nearest-candidate matching is part of the training objective. The latent dimension is set to the action dimension, $d_z = d_a$, so the actor input noise has the same dimensionality as the action it generates. For the main DROL and FQL results, reported means and standard deviations are computed over eight training seeds, 1996–2003, unless a table entry is copied from a baseline source that reports only a single number. The supplementary candidate ablation uses four seeds and the mechanism-training diagnostics use two, as specified in Section E. Seeds are fixed in advance, and evaluation uses the final checkpoint. For grouped OGBench rows, we first average the five task scores within each seed and then compute the cross-seed mean and standard deviation. Experiments were run on NVIDIA RTX 3090 and NVIDIA A100 GPUs; the runtime measurements in Figure 3 were collected on an NVIDIA A100. The experiments use the public OGBench and D4RL benchmark datasets and baseline implementations under their public release terms.

Table 2: Shared architecture and optimization settings used by DROL unless overridden by benchmark-specific flags.

Setting	Value
Optimizer	Adam, learning rate 3×10^{-4}
Batch size	256
Actor / critic	4-layer MLPs with hidden widths $(512, 512, 512, 512)$
Critic ensemble	2
Critic target aggregation	mean by default; min with `q_agg=min`
Actor Q aggregation	ensemble minimum
Target update rate τ	0.005
Default discount γ	0.99
Default routing budget K	16
Offline steps	10^6 on OGBench, 5×10^5 on D4RL
Evaluation	every 100,000 updates, 50 episodes

For OGBench, `DROL(16)` keeps the default $K = 16$ in every family. We select α with the same family-level search scope and budget as FQL, reusing its selected setting in most families. Only after freezing α do we sweep K; this additional search belongs to the budget of `DROL`*. For `DROL`*, we choose one family-level routing budget $K^\star$ on the representative task marked by `(*)` in Table 4 and reuse it for the other tasks in that family without changing any other hyperparameters. The reported puzzle-3x3 entries retain the fixed-$K = 16$ reference in both columns; they are not results from $K = 32$. For D4RL, all runs use 500K offline updates. The `DROL`* protocol is benchmark-specific. On D4RL Adroit, we sweep K with the natural task families in mind (for example, `pen-*`, `door-*`, `hammer-*`, and `relocate-*`) and report the resulting choices in Table 3. On D4RL AntMaze, the reported table keeps the default $K = 16$ for all tasks after selecting the task-level α values shown below; the additional K sweeps in Figure 4 are diagnostic rather than a separate reported `DROL`* protocol. The table below summarizes the benchmark-level (α, K) settings. The right column lists changes in hyperparameters, not differences in reported scores: a dash means that the listed settings are unchanged. Identical settings do not necessarily denote the same run set. For example, the AntMaze-Large and cube-single columns come from separate fixed-budget and sweep run sets, both with $K = 16$; their score differences are not effects of changing K. `q_agg=min` denotes taking the minimum over the two target critics when forming the TD target; without this flag, target critics are averaged.

B Related Work

Classical offline RL methods such as CQL (Kumar et al., 2020), TD3+BC (Fujimoto & Gu, 2021), and IQL (Kostrikov et al., 2022) control distribution shift mainly through critic-side pessimism or simple behavior regularization. As multimodal benchmarks such as OGBench have made actor

expressiveness more important, recent work has increasingly turned to diffusion and flow policies, including Diffusion-QL (Wang et al., 2023), IDQL (Hansen-Estruch et al., 2023), and flow-matching-based generators built on Lipman et al. (2023).

DROL is most closely related to generative offline RL methods that focus on efficient one-step policy learning once the critic is fixed or largely standard. FQL (Park et al., 2025b) is the clearest reference point: it learns an expressive flow teacher and distills it into a one-step actor for critic-guided optimization. MeanFlow-QL (Wang et al., 2025), SORL (Espinosa-Dice et al., 2025), FAC (Chae et al., 2026), and DeFlow (Mu et al., 2026) pursue adjacent efficiency or refinement strategies. Our pointwise-pairing comparison specifically concerns FQL's teacher–student objective; these other approaches need not share that objective. DROL instead trains its current candidate set directly against dataset actions, using the nearest match to select the recipient of both reconstruction and critic guidance.

Another closely related direction emphasizes support-aware flow design. ReFORM (Zhang et al., 2026) is especially relevant because it also argues that support, rather than generic distributional closeness, should be the operative constraint in offline RL, and it uses bounded and reflected flow constructions to preserve support by design. DROL is complementary in spirit but different in mechanism: rather than constraining policy improvement through a reflected flow transformation, it uses dynamic routing over a one-step candidate set and lets responsibility move between candidates as the local partition changes.

At the policy-learning level, the routed behavior-cloning term is most closely related to IMLE (Li & Malik, 2019) and to its policy instantiation in IMLE Policy (Rana et al., 2025). These methods train a conditional generator through nearest-sample or set-level matching rather than pointwise sample-to-target correspondence, which is why they are the right comparison point for DROL's winner-only BC objective. The novelty here is not the matching primitive itself, but its use as the support-aware regularizer coupled directly to local Q-improvement on the routed winner. DROL also differs from IBC (Florence et al., 2022), which models the policy implicitly with an energy-based objective rather than a direct latent-conditioned generator with hard routing during training.

Table 3: Benchmark-level hyperparameters. Each pair is (α, K).

Dataset / family	`DROL(16)`	`DROL`* if different	Extra flags
OGBench			
`antmaze-giant`	$(0.03, 16)$	$(0.03, 1)$	`discount=0.995`, `q_agg=min`
`antmaze-large`	$(0.1, 16)$	–	`discount=0.995`, `q_agg=min`
`antsoccer-arena`	$(0.1, 16)$	$(0.1, 4)$	`discount=0.995`
`humanoidmaze-medium`	$(0.3, 16)$	–	`discount=0.995`
`humanoidmaze-large`	$(0.3, 16)$	$(0.3, 4)$	`discount=0.995`
`cube-single`	$(10.0, 16)$	–	–
`cube-double`	$(3.0, 16)$	$(3.0, 64)$	–
`scene`	$(3.0, 16)$	$(3.0, 64)$	–
`puzzle-3x3`	$(0.3, 16)$	–	–
`puzzle-4x4`	$(10.0, 16)$	$(10.0, 64)$	–
D4RL			
`antmaze-umaze`	$(0.3, 16)$	–	–
`antmaze-umaze-diverse`	$(0.1, 16)$	–	–
`antmaze-medium-play`	$(0.1, 16)$	–	–
`antmaze-medium-diverse`	$(0.1, 16)$	–	–
`antmaze-large-play`	$(0.03, 16)$	–	–
`antmaze-large-diverse`	$(0.03, 16)$	–	–
`pen`	$(1.0, 16)$	–	`q_agg=min`
`door`	$(10.0, 16)$	$(10.0, 8)$	`q_agg=min`
`hammer`	$(3.0, 16)$	$(3.0, 8)$	`q_agg=min`
`relocate`	$(10.0, 16)$	$(10.0, 32)$	`q_agg=min`

C Proof Details

This appendix records proofs for the propositions in Section 4. The goal is to make the mechanism-level claims explicit. We do not attempt a full convergence theory for offline RL.

C.1 Separated Neighborhoods Avoid Collapse

We prove a global distortion bound and characterize equality, allowing Voronoi cells to cross gaps and allowing an interval to be split between several cells. Write $x_i = \hat{a}_i$ and sort the M prototypes in increasing order.

Compressing the gaps. Let $L = 2rM$ and define the increasing map $T : [0, L] \to \bigcup_{m=1}^{M} I_m$, up to endpoints of measure zero, by

$$T(t) = c_m - r + t - 2r(m-1), \qquad 2r(m-1) \le t < 2rm.$$

The pushforward of $\mathrm{Unif}([0, L])$ under T is exactly ρ, since all intervals have the same length and mixture weight. The map has slope one within each interval and positive jumps across the gaps. Consequently,

$$|T(u) - T(v)| \ge |u - v| \qquad \text{for almost every } u, v \in [0, L].$$

A bound for every Voronoi cell. For distinct ordered prototypes, their one-dimensional nearest-neighbor cells are intervals, with ties at isolated boundaries. Their preimages under T therefore form an ordered partition $J_1, \dots, J_M$ of $[0, L]$ into intervals, some possibly empty. For duplicate prototypes, assign tied points to a single representative; the duplicate cells can be empty, so the same argument applies. Write $\ell_i = |J_i|$, so $\sum_i \ell_i = L$. For $\ell_i > 0$, define $\bar{T}_i = \ell_i^{-1} \int_{J_i} T(t)\,\mathrm{d}t$. The variance decomposition and the pairwise representation of variance give

$$\begin{aligned}
\int_{J_i} (T(t) - x_i)^2\,\mathrm{d}t &= \int_{J_i} (T(t) - \bar{T}_i)^2\,\mathrm{d}t + \ell_i (x_i - \bar{T}_i)^2 \\
&\ge \frac{1}{2\ell_i} \int_{J_i}\int_{J_i} (T(u) - T(v))^2\,\mathrm{d}u\,\mathrm{d}v \\
&\ge \frac{1}{2\ell_i} \int_{J_i}\int_{J_i} (u - v)^2\,\mathrm{d}u\,\mathrm{d}v = \frac{\ell_i^3}{12}.
\end{aligned}$$

An empty cell contributes zero, consistent with the same bound. Summing and applying Jensen's inequality yields

$$\mathcal{J}(x_{1:M}) = \frac{1}{L} \sum_{i=1}^{M} \int_{J_i} (T(t) - x_i)^2\,\mathrm{d}t \ge \frac{1}{12L} \sum_{i=1}^{M} \ell_i^3 \ge \frac{M}{12L} \left(\frac{L}{M}\right)^3 = \frac{r^2}{3}.$$

This bound does not assume that any original interval already has its own prototype.

Attainment and equality. Placing $x_m = c_m$ attains $r^2/3$: the midpoint between consecutive centers lies in the separating gap, so each I_m is assigned to c_m and contributes its uniform variance $r^2/3$ before mixture weighting. Hence a global minimizer exists and has this distortion. At any global minimizer, equality in the strictly convex Jensen bound forces $\ell_i = 2r$ for every i; in particular, no cell is empty. Because the J_i are an ordered interval partition starting at zero, they must be $[2r(i-1), 2ri]$ up to null endpoints. Thus $T(J_i) = I_i$ up to null sets. Equality in the variance decomposition further requires $x_i = \bar{T}_i = c_i$. Every global minimizer therefore places exactly one prototype at each interval center, up to permutation, proving the proposition.

The proof uses the equal lengths and equal mixture weights essentially. It requires only positive gaps ($c_{m+1} - c_m > 2r$), so it also covers the stronger separation assumed in the proposition; no analogous claim for unequal masses or a shared neural actor follows from this argument.

C.2 Persistent Tether versus Routed Anchor

For the pointwise-correspondence loss,

$$\ell_{\mathrm{fix}}(x; a) = \|x - a\|_2^2 - \alpha Q_\phi(s, x),$$

the gradient is obtained directly by differentiation:

$$\nabla_x \ell_{\mathrm{fix}}(x; a) = 2(x - a) - \alpha \nabla_x Q_\phi(s, x).$$

The pullback term $2(x - a)$ is always attached to the same variable x.

For the routed loss,

$$\ell_{\mathrm{route}}(A; a) = \min_{k \in [K]} \|x_k - a\|_2^2 - \alpha Q_\phi(s, x_{k^*(a)}),$$

fix a point where the winner $k^*(a)$ is unique and write $k^* = k^*(a)$. Define the routing margin

$$\delta := \min_{j \neq k^*} \Big(\|x_j - a\|_2^2 - \|x_{k^*} - a\|_2^2 \Big) > 0.$$

By continuity of squared distance, there is a neighborhood of A in which this strict ordering is preserved, so the winner remains k^* throughout that neighborhood. On that neighborhood,

$$\ell_{\mathrm{route}}(A; a) = \|x_{k^*} - a\|_2^2 - \alpha Q_\phi(s, x_{k^*}).$$

Hence the loss is locally independent of every non-winner x_i with $i \neq k^*$, which gives

$$\nabla_{x_i} \ell_{\mathrm{route}}(A; a) = 0, \qquad i \neq k^*,$$

and

$$\nabla_{x_{k^*}} \ell_{\mathrm{route}}(A; a) = 2(x_{k^*} - a) - \alpha \nabla_x Q_\phi(s, x_{k^*}).$$

When the nearest candidate changes, the winner index changes as well, so the same behavior-cloning term is attached to a different prototype.

C.3 Fixed Correspondence Biases a Local Improvement

Let

$$g(x) = \|x - a\|_2^2 - \alpha Q_\phi(s, x)$$

on the closed convex set U. Since $Q_\phi(s, \cdot)$ is m-strongly concave, the function $-Q_\phi(s, \cdot)$ is m-strongly convex. The quadratic term $\|x - a\|_2^2$ is 2-strongly convex, so g is $(2 + \alpha m)$-strongly convex on U. Strong convexity on the closed convex set U implies that g has a unique minimizer x_{fix}.

Because x_Q is an interior maximizer of $Q_\phi(s, \cdot)$, we have $\nabla_x Q_\phi(s, x_Q) = 0$. Therefore

$$\nabla g(x_Q) = 2(x_Q - a).$$

This vanishes if and only if $a = x_Q$, which proves the first claim.

For the second claim, use strong monotonicity of ∇g:

$$\langle \nabla g(x_{\mathrm{fix}}) - \nabla g(x_Q),\, x_{\mathrm{fix}} - x_Q \rangle \geq (2 + \alpha m) \|x_{\mathrm{fix}} - x_Q\|_2^2.$$

The constrained first-order optimality condition gives

$$\langle \nabla g(x_{\mathrm{fix}}),\, x_Q - x_{\mathrm{fix}} \rangle \geq 0,$$

or equivalently

$$\langle \nabla g(x_{\mathrm{fix}}),\, x_{\mathrm{fix}} - x_Q \rangle \leq 0.$$

Since $\nabla g(x_Q) = 2(x_Q - a)$, the previous display implies

$$-2\langle x_Q - a,\, x_{\mathrm{fix}} - x_Q \rangle \geq (2 + \alpha m) \|x_{\mathrm{fix}} - x_Q\|_2^2.$$

Applying Cauchy–Schwarz and canceling one factor of $\|x_{\mathrm{fix}} - x_Q\|_2$ yields

$$\|x_{\mathrm{fix}} - x_Q\|_2 \leq \frac{2}{2 + \alpha m} \|a - x_Q\|_2.$$

The first claim, together with the interiority of x_Q, shows that $x_{\mathrm{fix}} \neq x_Q$ whenever $a \neq x_Q$. The displayed norm inequality is an upper bound on the displacement, not a positive lower bound. Before applying Cauchy–Schwarz, the inequality also gives

$$\langle a - x_Q,\, x_{\mathrm{fix}} - x_Q \rangle \geq \frac{2 + \alpha m}{2} \|x_{\mathrm{fix}} - x_Q\|_2^2,$$

so a nonzero displacement has a positive projection toward the anchor, without requiring the two vectors to be parallel.

C.4 Coverage from Larger Candidate Sets

Fix s and a modal neighborhood $U_m(s)$. For each m, define the miss event

$$E_m := \{\hat{a}_k \notin U_m(s) \text{ for all } k \in [K]\}.$$

Because the K latent samples are independent and each single sample lands in $U_m(s)$ with probability $p_m(s)$,

$$\mathbb{P}(E_m) = (1 - p_m(s))^K.$$

The event that some neighborhood is missed is $\bigcup_{m=1}^{M_s} E_m$, so a union bound gives

$$\mathbb{P}\Big(\exists m \text{ such that all } K \text{ candidates miss } U_m(s)\Big) \le \sum_{m=1}^{M_s} (1 - p_m(s))^K.$$

Taking complements yields the first inequality in the proposition. For the exponential form, use $(1-p)^K \le e^{-Kp}$ together with $p_m(s) \ge p_{\min}(s)$:

$$\sum_{m=1}^{M_s} (1 - p_m(s))^K \le \sum_{m=1}^{M_s} e^{-K p_m(s)} \le M_s e^{-K p_{\min}(s)}.$$

Substituting this into the previous display gives the stated bound.

C.5 Scale of the Bounded Latent Prior

Let $z \sim \mathrm{Unif}(B_R^{d_z})$. By rotational symmetry, the radius $r = \|z\|_2$ has density

$$f(r) = \frac{d_z\, r^{d_z - 1}}{R^{d_z}}, \qquad 0 \le r \le R.$$

Therefore

$$\mathbb{E}\|z\|_2^2 = \int_0^R r^2 f(r)\, \mathrm{d}r = \frac{d_z}{R^{d_z}} \int_0^R r^{d_z+1}\, \mathrm{d}r = \frac{d_z}{d_z + 2} R^2.$$

With the ReFORM choice $R = \sqrt{d_a}$, this becomes

$$\mathbb{E}\|z\|_2^2 = \frac{d_z}{d_z + 2} d_a,$$

so the overall latent scale is of the same order as the action dimension.

D Additional OGBench and D4RL Results

This appendix reports the full task-level benchmark breakdown corresponding to the grouped OGBench and D4RL summaries in the main text.

Table 4: **Full task-level benchmark layout on OGBench and D4RL.** DROL(16) uses the fixed default $K = 16$. DROL* uses the benchmark-specific K-selection protocol described in Appendix A. Bold entries are at least $0.95\times$ the best score in their row, and underlined entries are one-step methods that are at least $0.95\times$ the best score.

	Gaussian Policies			Multi-step / Iterative Policies†					One-step Methods		
Task	BC	IQL	ReBRAC	FAWAC	FBRAC	IFQL	DeFlow	SORL	FQL	DROL(16)	DROL*
antmaze-large-navigate-singletask-task1-v0 (*)	0 ±0	48 ±9	91 ±10	1 ±1	70 ±20	24 ±17	88 ±6	93 ±2	80 ±8	**98** ±2	**98** ±2
antmaze-large-navigate-singletask-task2-v0	6 ±3	42 ±6	**88** ±4	0 ±1	35 ±12	8 ±3	71 ±10	79 ±5	57 ±10	**86** ±7	**84** ±6
antmaze-large-navigate-singletask-task3-v0	29 ±5	72 ±7	51 ±18	12 ±4	83 ±15	52 ±17	88 ±5	88 ±10	**93** ±3	88 ±5	**90** ±5
antmaze-large-navigate-singletask-task4-v0	8 ±3	51 ±9	84 ±7	10 ±3	37 ±18	18 ±8	77 ±5	**91** ±2	80 ±4	84 ±8	79 ±14
antmaze-large-navigate-singletask-task5-v0	10 ±3	54 ±22	90 ±2	9 ±5	76 ±8	38 ±18	84 ±3	**95** ±0	83 ±4	**95** ±4	**93** ±3
antmaze-giant-navigate-singletask-task1-v0 (*)	0 ±0	0 ±0	**27** ±22	0 ±0	0 ±1	0 ±0	4 ±4	12 ±6	4 ±5	0 ±0	**28** ±31
antmaze-giant-navigate-singletask-task2-v0	0 ±0	1 ±1	16 ±17	0 ±0	4 ±7	0 ±0	36 ±13	0 ±0	9 ±7	0 ±0	**54** ±39
antmaze-giant-navigate-singletask-task3-v0	0 ±0	0 ±0	34 ±22	0 ±0	0 ±0	0 ±0	2 ±3	0 ±0	0 ±1	0 ±0	**77** ±9
antmaze-giant-navigate-singletask-task4-v0	0 ±0	0 ±0	5 ±12	0 ±0	9 ±4	0 ±0	5 ±8	**25** ±18	14 ±23	0 ±0	8 ±22
antmaze-giant-navigate-singletask-task5-v0	1 ±1	19 ±7	49 ±22	0 ±0	6 ±10	13 ±9	16 ±19	6 ±15	16 ±28	13 ±17	**58** ±26
humanoidmaze-medium-navigate-singletask-task1-v0 (*)	1 ±0	32 ±7	16 ±9	6 ±2	25 ±8	**69** ±19	21 ±11	**67** ±4	19 ±12	58 ±15	55 ±16
humanoidmaze-medium-navigate-singletask-task2-v0	1 ±0	41 ±9	18 ±16	40 ±2	76 ±10	85 ±11	69 ±5	89 ±3	**94** ±3	88 ±24	**95** ±11
humanoidmaze-medium-navigate-singletask-task3-v0	6 ±2	25 ±5	36 ±13	19 ±2	27 ±11	49 ±49	63 ±14	**83** ±4	74 ±18	78 ±25	**79** ±20
humanoidmaze-medium-navigate-singletask-task4-v0	0 ±0	0 ±1	15 ±16	1 ±1	1 ±2	1 ±1	6 ±7	1 ±0	3 ±4	15 ±8	**17** ±10
humanoidmaze-medium-navigate-singletask-task5-v0	2 ±1	66 ±4	24 ±20	31 ±7	63 ±9	**98** ±2	81 ±7	81 ±20	**97** ±2	84 ±24	**96** ±18
humanoidmaze-large-navigate-singletask-task1-v0 (*)	0 ±0	3 ±1	2 ±1	0 ±0	0 ±1	6 ±2	3 ±4	**20** ±9	7 ±6	**21** ±22	**20** ±15
humanoidmaze-large-navigate-singletask-task2-v0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±4	**0** ±0	**0** ±0	**0** ±0	**0** ±0
humanoidmaze-large-navigate-singletask-task3-v0	1 ±1	7 ±3	8 ±4	1 ±1	10 ±2	**48** ±10	10 ±6	5 ±2	11 ±7	16 ±7	34 ±14
humanoidmaze-large-navigate-singletask-task4-v0	1 ±0	1 ±0	1 ±1	0 ±0	0 ±0	1 ±1	**8** ±3	0 ±0	2 ±3	0 ±0	1 ±1
humanoidmaze-large-navigate-singletask-task5-v0	0 ±1	1 ±1	**2** ±2	0 ±0	1 ±1	0 ±0	1 ±1	0 ±0	1 ±3	0 ±0	0 ±0
antsoccer-arena-navigate-singletask-task1-v0	2 ±1	14 ±5	0 ±0	22 ±2	17 ±3	61 ±25	84 ±7	**93** ±4	77 ±4	80 ±6	76 ±6
antsoccer-arena-navigate-singletask-task2-v0	2 ±2	17 ±7	0 ±1	8 ±1	8 ±2	75 ±3	89 ±4	**96** ±2	88 ±3	**93** ±4	87 ±7
antsoccer-arena-navigate-singletask-task3-v0	0 ±0	6 ±4	0 ±0	11 ±5	16 ±3	14 ±22	56 ±7	55 ±6	**61** ±6	**58** ±3	**60** ±7
antsoccer-arena-navigate-singletask-task4-v0 (*)	1 ±0	3 ±2	0 ±0	12 ±3	24 ±4	16 ±9	43 ±5	**54** ±5	39 ±6	39 ±9	41 ±7
antsoccer-arena-navigate-singletask-task5-v0	0 ±0	2 ±2	0 ±0	9 ±2	15 ±4	0 ±1	39 ±9	47 ±9	36 ±9	42 ±12	**60** ±8
cube-single-play-singletask-task1-v0	10 ±5	88 ±3	89 ±5	81 ±9	73 ±33	79 ±4	**96** ±3	**97** ±2	**97** ±2	**97** ±2	**98** ±2
cube-single-play-singletask-task2-v0 (*)	3 ±1	85 ±8	92 ±4	81 ±9	83 ±13	73 ±3	**99** ±1	**99** ±0	**97** ±2	**99** ±1	**99** ±1
cube-single-play-singletask-task3-v0	9 ±3	91 ±5	93 ±3	87 ±4	82 ±12	88 ±4	**100** ±1	**99** ±1	**98** ±2	**99** ±1	**100** ±1
cube-single-play-singletask-task4-v0	2 ±1	73 ±6	**92** ±3	79 ±6	79 ±20	79 ±6	**95** ±4	**95** ±2	**94** ±3	**95** ±3	**96** ±2
cube-single-play-singletask-task5-v0	3 ±3	78 ±9	87 ±8	78 ±10	76 ±33	77 ±7	**92** ±5	**93** ±3	**93** ±3	**94** ±5	**96** ±2
cube-double-play-singletask-task1-v0	8 ±3	27 ±5	45 ±6	21 ±7	47 ±11	35 ±9	65 ±6	**77** ±11	61 ±9	65 ±7	65 ±14
cube-double-play-singletask-task2-v0 (*)	0 ±0	1 ±1	7 ±3	2 ±1	22 ±12	9 ±5	**49** ±13	33 ±8	36 ±6	22 ±8	28 ±7
cube-double-play-singletask-task3-v0	0 ±0	0 ±0	4 ±1	1 ±1	4 ±2	8 ±5	**34** ±15	12 ±6	22 ±5	9 ±5	22 ±6
cube-double-play-singletask-task4-v0	0 ±0	0 ±0	1 ±1	0 ±0	0 ±1	1 ±1	7 ±4	7 ±4	5 ±2	2 ±2	**9** ±4
cube-double-play-singletask-task5-v0	0 ±0	4 ±3	4 ±2	2 ±1	2 ±2	17 ±6	**46** ±11	1 ±1	19 ±10	2 ±2	8 ±2
scene-play-singletask-task1-v0	19 ±6	94 ±3	**95** ±2	87 ±8	**96** ±8	**98** ±3	**98** ±2	**99** ±1	**100** ±0	**100** ±0	**100** ±0
scene-play-singletask-task2-v0 (*)	1 ±1	12 ±3	50 ±13	18 ±8	46 ±10	0 ±0	59 ±17	**89** ±9	76 ±9	69 ±12	**89** ±8
scene-play-singletask-task3-v0	1 ±1	32 ±7	55 ±16	38 ±9	78 ±14	54 ±19	85 ±5	**97** ±1	**98** ±1	**98** ±2	**98** ±2
scene-play-singletask-task4-v0	2 ±2	0 ±1	3 ±3	6 ±1	4 ±4	0 ±0	**13** ±8	1 ±1	5 ±1	3 ±4	3 ±2
scene-play-singletask-task5-v0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0
puzzle-3x3-play-singletask-task1-v0	5 ±2	33 ±6	**97** ±4	25 ±9	63 ±19	**94** ±3	89 ±7	–	90 ±4	**93** ±9	**93** ±9
puzzle-3x3-play-singletask-task2-v0	1 ±1	4 ±3	1 ±1	4 ±2	2 ±2	1 ±2	**38** ±11	–	16 ±5	24 ±31	24 ±31
puzzle-3x3-play-singletask-task3-v0	1 ±1	3 ±2	3 ±1	1 ±0	1 ±1	0 ±0	**35** ±11	–	10 ±3	7 ±4	7 ±4
puzzle-3x3-play-singletask-task4-v0 (*)	1 ±1	2 ±1	2 ±1	1 ±1	2 ±2	0 ±0	23 ±8	–	16 ±5	**36** ±19	**36** ±19
puzzle-3x3-play-singletask-task5-v0	1 ±0	3 ±2	5 ±3	1 ±1	2 ±2	0 ±0	31 ±8	–	16 ±3	**46** ±23	**46** ±23
puzzle-4x4-play-singletask-task1-v0	1 ±1	12 ±2	26 ±4	1 ±2	32 ±9	**49** ±9	23 ±3	–	34 ±8	28 ±6	39 ±6
puzzle-4x4-play-singletask-task2-v0	0 ±0	7 ±4	12 ±4	0 ±1	5 ±3	4 ±4	9 ±2	–	16 ±5	11 ±3	**20** ±7
puzzle-4x4-play-singletask-task3-v0	0 ±0	9 ±3	15 ±3	1 ±1	20 ±10	**50** ±14	9 ±4	–	18 ±5	18 ±7	32 ±8
puzzle-4x4-play-singletask-task4-v0 (*)	0 ±0	5 ±2	10 ±3	0 ±0	5 ±1	**21** ±11	7 ±4	–	11 ±3	12 ±4	16 ±4
puzzle-4x4-play-singletask-task5-v0	0 ±0	4 ±1	7 ±3	0 ±1	4 ±3	2 ±2	7 ±4	–	7 ±3	12 ±3	**17** ±4
antmaze-umaze-v2	55	77	**98**	90 ±6	**94** ±3	92 ±6	**96** ±3	–	**96** ±2	**94** ±2	**94** ±2
antmaze-umaze-diverse-v2	47	54	84	55 ±7	82 ±9	62 ±12	**87** ±8	–	**89** ±5	**86** ±8	**86** ±8
antmaze-medium-play-v2	0	66	**90**	52 ±12	77 ±7	56 ±15	80 ±7	–	78 ±7	84 ±7	84 ±7
antmaze-medium-diverse-v2	1	74	**84**	44 ±15	77 ±6	60 ±25	76 ±19	–	71 ±13	72 ±11	72 ±11
antmaze-large-play-v2	0	42	52	10 ±6	32 ±21	55 ±9	78 ±6	–	**84** ±7	72 ±4	72 ±4
antmaze-large-diverse-v2	0	30	64	16 ±10	20 ±17	64 ±8	78 ±6	–	**83** ±4	76 ±4	76 ±4
pen-human-v1	71	78	**103**	67 ±5	77 ±7	71 ±12	61 ±9	–	53 ±6	52 ±9	52 ±4
pen-cloned-v1	52	83	**103**	62 ±10	67 ±9	80 ±11	75 ±4	–	74 ±11	78 ±12	78 ±12
pen-expert-v1	110	128	**152**	118 ±6	119 ±7	139 ±5	138 ±5	–	142 ±6	137 ±6	137 ±6
door-human-v1	2	3	0	2 ±1	4 ±2	**7** ±2	1 ±0	–	0 ±0	0 ±0	0 ±0
door-cloned-v1	0	**3**	0	0 ±1	0 ±0	2 ±2	2 ±1	–	2 ±1	1 ±1	1 ±1
door-expert-v1	**105**	**107**	**106**	**103** ±1	**104** ±1	**104** ±2	**104** ±1	–	**104** ±1	**105** ±1	**105** ±0
hammer-human-v1	**3**	2	0	2 ±1	2 ±1	**3** ±1	1 ±0	–	1 ±1	2 ±2	2 ±2
hammer-cloned-v1	1	2	5	1 ±0	2 ±1	2 ±1	7 ±4	–	**11** ±9	8 ±3	8 ±3
hammer-expert-v1	127	**129**	**134**	118 ±3	119 ±9	117 ±9	127 ±1	–	125 ±3	**128** ±1	**128** ±1
relocate-human-v1	**0**	**0**	**0**	**0** ±0	**0** ±0	**0** ±0	**0** ±0	–	**0** ±0	**0** ±0	**0** ±0
relocate-cloned-v1	0	0	**2**	0 ±0	1 ±1	0 ±0	1 ±8	–	0 ±0	0 ±0	0 ±0
relocate-expert-v1	**108**	**106**	**108**	**105** ±3	**105** ±2	**104** ±3	**109** ±2	–	**107** ±1	**107** ±2	**108** ±1

E Candidate Ablations and Benchmark Mechanism Diagnostics

E.1 Matched Candidate-Budget Ablation

We compare $K = 1$ and $K = 16$ on all five tasks of cube-double, Scene, and puzzle-4x4. Both configurations use seeds 1996–1999, 1M offline updates, final-checkpoint evaluation with 50 episodes, and the same actor, bounded prior, discount, family hyperparameters, and α. Only K changes. Within each seed, we average the five task success rates, then compute the cross-seed mean and sample standard deviation. The paired difference is computed between these family means. These four-seed comparisons are separate from the eight-seed headline table.

Table 5: Complete-family candidate ablation. Scores are success percentages, mean ± sample SD across four seeds. Intervals are paired percentile-bootstrap intervals across the four seed differences.

Family	$K = 1$	$K = 16$	Difference	95% bootstrap CI
Cube-double	1.8 ± 0.8	18.8 ± 3.4	$+17.0$	$[14.8, 19.2]$
Scene	32.3 ± 4.7	53.5 ± 3.5	$+21.2$	$[16.8, 27.8]$
Puzzle-4x4	1.8 ± 1.2	13.4 ± 2.8	$+11.6$	$[9.0, 14.0]$

All four seed differences favor $K = 16$ within each family. The small sample limits inferential precision: the exact two-sided sign-flip test yields $p = 0.125$ for each family. The bootstrap intervals describe the observed effect size but should not be read as evidence that these exact tests reject at the 5% level. These comparisons test the combined effect of changing candidate matching and the actions selected for Q updates.

E.2 Final-Checkpoint Geometry

We use seeds 1996–1999 and 4,096 fixed offline transitions per seed on cube-double task 2, Scene task 2, and puzzle-4x4 task 4. Distances are per-action-dimension RMSE to the paired dataset action. A near tie means that the second-best squared distance is within 10% of the best. The same candidate bank supplies the routed and single-draw measurements within a run.

Table 6: Final-checkpoint geometry, averaged over four seeds. Distances are paired-action proxies; they are not tests of membership in the full behavior support.

Task	K	Routed RMSE	Single-draw RMSE	Near ties
Cube-double task 2	16	0.062	0.141	19.6%
	64	0.052	0.163	21.0%
Scene task 2	16	0.064	0.152	21.1%
	64	0.055	0.177	21.6%
Puzzle-4x4 task 4	16	0.058	0.138	19.6%
	64	0.047	0.150	20.7%

The nearest candidate is closer at larger K, while the random single draw is farther from the paired action on all three tasks. The former combines a candidate-count effect with changes in the trained actor. The latter need not imply unsupported behavior: a different action may belong to another valid mode. Neither statistic alone establishes support preservation.

E.3 Gradient Conflict in FQL

We train the released FQL implementation on Scene task 2 and AntMaze-Large task 1 with its released hyperparameters and seeds 1996–1997, for 1M updates without policy evaluation. Checkpoints are saved at 250k, 500k, 750k, and 1M. Each seed uses the same 4,096 fixed transitions across checkpoints. At actor output x and the corresponding flow-teacher endpoint $\tilde{a}$, define

$$g_Q = \nabla_x Q(s, x), \qquad g_T = \tilde{a} - x. \tag{9}$$

Output conflict is $g_Q^\top g_T < 0$. Separately, we compute the cosine between the mini-batch Q-loss and distillation-loss gradients restricted to the one-step actor parameters, on 16 fixed batches of size 256.

Table 7: FQL output-direction conflict. Entries are the mean across two seeds at each checkpoint. The actor-parameter gradients conflict on every fixed batch for both seeds at every checkpoint.

Task	250k	500k	750k	1M
AntMaze-Large task 1	69.5%	71.1%	71.9%	71.0%
Scene task 2	83.5%	81.7%	82.0%	82.0%

This audit documents interference in benchmark-trained FQL actors. It does not measure interference across the entire actor–critic system, and it does not establish that a negative cosine is harmful in every state.

E.4 Fixed-Probe Responsibility Reassignment

We train DROL with $K = 16$ on Scene task 2 using the submitted hyperparameters and two fixed seeds, 1996–1997, for 1M updates without policy evaluation. We save initialization and checkpoints every 50k updates. For each seed, the probe consists of 4,096 fixed offline targets and a fixed bank of 16 latent draws per target, reused across checkpoints. Thus candidate identities refer to fixed latent-conditioned outputs. Primary statistics use the 19 intervals after the first 50k checkpoint, excluding the random-initialization transient.

Table 8: Fixed-probe responsibility changes. Brackets give the range across the two training seeds, not confidence intervals.

Metric	Mean [seed range]
Targets with at least one owner change	99.8% [99.8%, 99.9%]
Targets with a change stable for 100k	76.1% [75.5%, 76.7%]
Owner changes per target	12.557 [12.534, 12.579]
Changes stable for the next two checkpoints	11.0% [10.9%, 11.1%]
Changes adjacent to a near tie	40.5% [40.4%, 40.6%]

A stable event keeps the new owner at the next two checkpoints; events in the last two checkpoints are excluded from that denominator. Near-tie adjacency means that either endpoint of a change has a relative squared-distance margin at most 10%. The target-level and event-level stability fractions have different denominators: many targets eventually experience a persistent change even though most individual changes are temporary. This is a fixed-probe checkpoint-scale diagnostic. Training resamples latents, so these counts are not literal per-update handoff frequencies. They establish that assignments evolve on real trained actors, but not that every reassignment improves return.