FREEZE, DON'T CLONE: FINE-TUNING WITH EXPRESSIVE POLICIES WITHOUT RETAINING OFFLINE DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline-to-online reinforcement learning uses offline pretraining to accelerate subsequent online learning. While many methods assume the offline dataset remains available throughout fine-tuning, there are many settings where this is not the case. Prior work attributes failures in this setting primarily to critic recalibration under distribution shift. We identify a complementary actor-side failure: once the offline data disappears, continued behavioral cloning (BC) no longer preserves the offline behavior distribution and instead refits the behavioral model to the agent's limited, and nonstationary, online experience. We demonstrate that stopping online behavioral fitting while preserving the pretrained expressive prior substantially improves no-retention fine-tuning across implicit flow Q-learning (IFQL), flow Q-learning (FQL), and action-chunked RL (QC) on five OGBench domains. We trace this failure to a behavioral feedback loop: continuing online behavioral cloning rapidly erases the pretrained prior, inducing representational collapse and action diversity loss that starves the critic of useful exploration. Freezing the prior breaks this cycle, preserving offline coverage while allowing value learning to drive continuous improvement. **Code is available in BC-Free Fine-Tuning.**

1 INTRODUCTION

Offline pretraining should allow an agent to learn from a dataset once and then improve through interaction without carrying that dataset forever. Yet most offline-to-online RL methods continue sampling offline transitions during online fine-tuning. This assumption can be expensive or infeasible when the original data are constrained by storage, privacy, licensing, or deployment requirements. Worse, simply deleting the offline buffer can cause an agent to rapidly lose the capabilities acquired during pretraining. We ask how an offline-pretrained agent should be fine-tuned when the original dataset is no longer available, which hereafter we refer to as "no-retention fine-tuning".

Previous work identifies the critic as an important source of instability in this setting. The transition to online interaction abruptly changes the state–action distribution, forcing value estimates learned from broad offline data to be recalibrated from a small online replay buffer. WSRL stabilizes this transition by collecting initial rollouts with the pretrained policy before beginning online updates (Zhou et al., 2025). This critic-centered view explains an important part of no-retention fine-tuning, but it leaves open what should happen to the behavioral model itself.

Behavioral regularization is central to many offline RL methods. By fitting or remaining close to dataset actions, the policy is discouraged from exploiting values assigned to unsupported actions (Kumar et al., 2019; Wu et al., 2019; Fujimoto & Gu, 2021; Tarasov et al., 2023). Recent approaches combine this principle with diffusion or flow-based policies that can represent complex and multi-modal behavior distributions (Hansen-Estruch et al., 2023; Park et al., 2025; Li et al., 2025). While the offline data remain available, continuing behavioral fitting can keep the policy anchored to the distribution learned during pretraining. When the offline dataset disappears, however, this regularization can become a liability. Behavioral cloning no longer preserves the offline distribution; it trains the behavioral model to imitate actions generated by the agent's current decision rule. Early online experience is scarce, correlated, policy-dependent, and nonstationary. Consequently, behaviors that are not immediately revisited may be undervalued and underused before improved value estimates can (re)highlight their value. This motivates a simple principle: *keep learning which actions are* valuable*, but stop relearning which actions are* possible.

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.

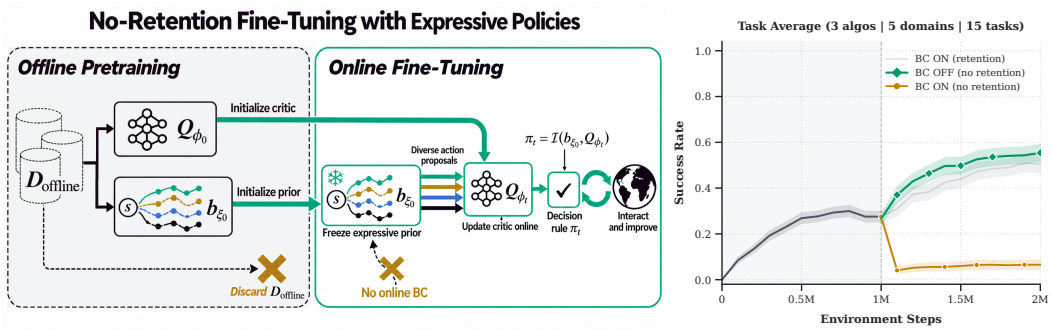

Figure 1: **Stopping BC enables no-retention fine-tuning. Left:** After offline pretraining, we freeze the behavioral prior and continue updating the critic using online data. **Right:** Aggregate performance across five OGBench tasks after 1M offline gradient steps (gray) and 1M online environment steps (white). Stopping BC consistently improves methods with expressive policies.

We instantiate this principle by stopping online behavior cloning (BC) to preserve the expressive prior learned offline. This does not freeze the entire agent: the critic and algorithm-specific improvement mechanisms continue learning from online experiences. IFQL makes this separation particularly clear: during the online stage a fixed flow policy proposes actions, while an updated critic improves which proposal is selected. In FQL, the flow policy remains fixed while the critic and a one-step actor continue learning, with the one step actor distilled toward the frozen flow policy. Action-chunked RL (QC) similarly preserves a prior over temporally extended behaviors while updating their values and selections.

Across five OGBench domains (15 tasks), preserving the pretrained prior substantially improves no-retention fine-tuning for IFQL, FQL, and QC. However, our ablations reveal that stopping BC alone is insufficient without an expressive architecture: the frozen prior must support complex, multimodal distributions to succeed. When the prior is restricted to a simple Gaussian, performance collapses. These results complement critic-centered accounts with a clear policy-side design principle: freeze an expressive, multimodal behavioral prior, and ensure online value learning can directly exploit it.

2 No-Retention Fine-Tuning with Expressive Priors

Problem Setting We consider a discounted Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ and seek a policy maximizing $J(\pi) = \mathbb{E}_{\pi,P}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. Offline pretraining uses $\mathcal{D}_{\text{off}} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^{N}$ to learn a critic Q_{ϕ_0} and a policy or behavioral model. During online fine-tuning, the agent accumulates $\mathcal{D}_{\text{on}}^t$ from interaction. We study the no-retention setting, in which $\mathcal{D}_{\text{off}}$ is unavailable after pretraining and all online updates use only $\mathcal{D}_{\text{on}}^t$. Let d_{off} and d_{on}^t denote the empirical state–action distributions of the two buffers. Retained-data fine-tuning may train on $d_t^\rho = \rho_t d_{\text{off}} + (1-\rho_t) d_{\text{on}}^t$ for $\rho_t > 0$. No retention sets $\rho_t = 0$. This change matters because a behavioral loss optimized under d_t^ρ remains partly supervised by offline behavior, whereas the same loss under d_{on}^t fits only the agent's evolving online experience.

Behavioral Priors and Decision Policies. We distinguish a *behavioral prior* $b_\xi(a \mid s)$ from the *decision policy* used for interaction. The prior models the conditional action distribution represented in its training data. For a likelihood-based model, behavioral fitting can be written as $\mathcal{L}_{\text{BC}}(\xi; \mathcal{D}) = -\mathbb{E}_{(s,a)\sim\mathcal{D}}[\log b_\xi(a \mid s)]$; deterministic, diffusion, and flow policies use the corresponding regression or generative objectives. The decision policy need not equal b_ξ. It may be represented by a separate value-optimized actor or constructed by critic-guided selection from the prior. We write this relationship as $\pi_t = \mathcal{I}_{\eta_t}(b_{\xi_t}, Q_{\phi_t})$, where $\mathcal{I}$ is a policy-improvement operator and η_t denotes any additional trainable parameters. Thus, fixing the prior need not fix the decision rule, which can still adapt through Q_{ϕ_t} or η_t.

Flow-Based Behavioral Priors A diagonal Gaussian is efficient but unimodal and may poorly model datasets containing several plausible actions at the same state. Flow policies instead transport noise $z \sim \mathcal{N}(0, I)$ to the behavior distribution using a state-conditioned velocity field (Liu et al., 2022; Lipman et al., 2023; Park et al., 2025). For a dataset action a, flow time $u \sim \mathcal{U}[0, 1]$, and interpolation

$x_u = (1-u)z + ua$, conditional flow matching minimizes $\mathcal{L}_{\mathrm{FM}}(\xi; \mathcal{D}) = \mathbb{E}_{(s,a),z,u}[\|f_\xi(s, x_u, u) - (a - z)\|_2^2]$. Integrating $\mathrm{d}x_u/\mathrm{d}u = f_\xi(s, x_u, u)$ from $x_0 = z$ to $u = 1$ produces $F_\xi(s, z)$, whose induced distribution defines $b_\xi(\cdot \mid s)$. Flow matching therefore plays the role of BC: it fits the behavioral prior to actions represented in the current replay data. IFQL uses this flow directly as a proposal distribution. In the best-of-K variant studied here, IFQL samples $a^{(i)} \sim b_\xi(\cdot \mid s)$ and executes $a^\star = \arg\max_i Q_\phi(s, a^{(i)})$. FQL instead retains F_ξ as a behavioral reference and trains a one-step actor $\mu_\psi(s, z)$ by minimizing $\mathcal{L}_{\mathrm{FQL}}(\psi) = \mathbb{E}_{s,z}[-Q_\phi(s, \mu_\psi(s, z)) + \alpha\|\mu_\psi(s, z) - \mathrm{sg}[F_\xi(s, z)]\|_2^2]$. Thus, FQL separates flow fitting from value-guided actor distillation.

Preserving the Prior During Online Fine-Tuning After offline pretraining, we freeze the behavioral prior, $\xi_t = \xi_0$, while updating the critic and algorithm-specific improvement parameters on $\mathcal{D}^t_{\mathrm{on}}$. The decision policy is therefore $\pi_t = \mathcal{I}_{\eta_t}(b_{\xi_0}, Q_{\phi_t})$. We denote continued behavioral fitting on online replay by BC-On and freezing the prior by BC-Off. Stopping BC affects only behavioral-model fitting; value learning, action selection, and distillation toward the frozen prior remain active. Under BC-Off, IFQL freezes its flow and updates the critic; FQL additionally updates its distilled one-step actor; and QC (Li et al., 2025) freezes its flow-based chunk prior while updating the chunk-level critic used for best-of-N selection.

Experimental Setup and Evaluation Protocol. Our primary benchmark spans five OGBench manipulation and locomotion domains (`cube-double`, `puzzle-4x4`, `antsoccer-arena`, `humanoidmaze-medium`, and `humanoidmaze-large`), evaluating three single-task variants per domain. For targeted extensions, we evaluate on low-quality data using `cube-double-noisy` and `antmaze-large-explore`, and examine human teleoperation demonstrations using the Multi-Human (`MH`) `Can` and `Lift` tasks from Robomimic (Mandlekar et al., 2021). Algorithms are pretrained for 1M gradient steps and fine-tuned for 1M environment steps; under no retention, the offline buffer is discarded at the transition. We compare BC-On versus BC-Off across IFQL, FQL, and QC, focusing detailed diagnostics on IFQL and FQL. All evaluations use three independent seeds per task; we report mean success rates with shaded standard error (SE) bands. Full environment specifications, task selection rationale, hyperparameters, and per-task results are detailed in Sec. B.2.

3 Freeze the Prior, Improve the Decision Rule

We first show that freezing the behavioral prior improves no-retention fine-tuning with flow-based policies. We then test whether removing BC is sufficient without an expressive prior and whether freezing remains effective after pretraining on low-quality data.

While our primary evaluations include QC, we focus our detailed diagnostic ablations on IFQL and FQL. In addition to their strong performance across recent offline-to-online RL literature (Park et al., 2025; Li & Levine, 2026), these two algorithms cleanly span the two fundamental mechanisms for utilizing an expressive prior: direct critic-guided action selection (IFQL) and actor distillation (FQL), while avoiding confounding factors from temporal abstraction.

Stopping BC improves no-retention fine-tuning. Fig. 3 compares BC-On and BC-Off during no-retention fine-tuning. Across IFQL, FQL, and action-chunked RL, stopping behavioral fitting consistently improves online learning. The largest gains occur for IFQL: BC-Off improves the aggregate mean and outperforms BC-On across tasks. FQL and the action-chunked methods exhibit the same overall trend, although the magnitude of the improvement varies across algorithms and domains. The IFQL and QC results are particularly revealing. Neither directly optimizes their behavioral flow (or a distilled version of it) for return; the flow proposes candidate actions, and the critic selects among them. Under BC-Off, the proposal distribution remains fixed while the critic continues learning from online experience. Performance can therefore improve without refitting the behavioral model. Under BC-On, the same flow matching loss is continually optimized on online-only replay, and performance is substantially lower. These results show that an objective useful during offline pretraining need not remain useful after its original data disappear. In this setting, preserving the pretrained behavioral model is more effective than continually refitting it to the agent's recent experience.

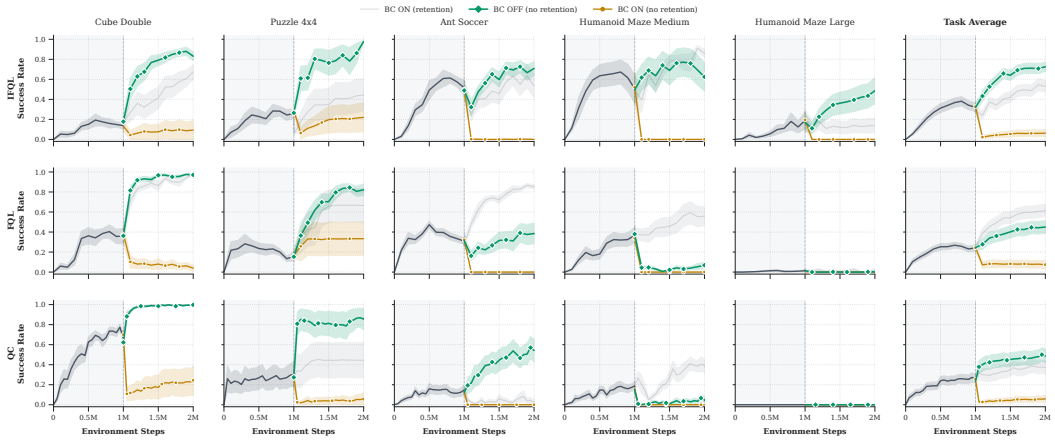

Figure 3: **Stopping BC improves no-retention fine-tuning across expressive policy classes.** Performance of IFQL, FQL, and QC (Action-Chunked RL) across five OGBench domains. The vertical dashed line marks the transition from offline pretraining to online fine-tuning. Continuing BC without offline data (BC-ON, orange dashed) causes catastrophic performance collapse. BC-OFF (red dashed) rescues online fine-tuning and often approaches the performance of the standard retained-data baseline (blue solid).

Attenuating the BC penalty cannot prevent collapse. To determine whether the failure of online behavioral cloning is merely an artifact of an overly aggressive regularization weight, we evaluate no-retention fine-tuning across varying BC penalty magnitudes ($0.1\times$ to $10\times$ the default). Fig. 2 demonstrates that scaling the BC weight, whether amplifying or aggressively attenuating it, fails to prevent catastrophic collapse in both IFQL and FQL. This confirms the failure mode is fundamentally tied to the objective itself rather than its hyperparameter tuning: any active optimization toward the agent's narrowing online experience progressively erases the offline prior. Consequently, only the complete removal of the behavioral loss (BC-OFF) enables sustained online improvement.

Can a large offline warmup rescue continued BC? Because the failure of BC-ON is directly tied to discarding offline data, we test whether a temporary data "warmup" can prevent the collapse. We pre-populate the online buffer using the pretrained checkpoint with 25,000 online transitions. As shown in (Fig. 4, Col 4 & Col 5), this intervention prevents the catastrophic failure of BC-ON; performance no longer flattens at near zero. However, across both IFQL and FQL on `cube-double` and `antsoccer-arena`, combining this warmup with BC-OFF still yields vastly superior online adaptation. This confirms that continued BC can fundamentally restrict the ceiling of policy improvement even when provided with healthy data initialization.

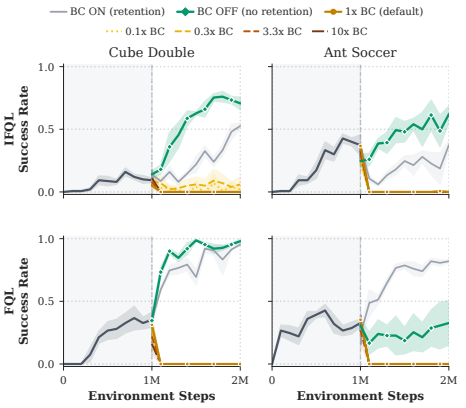

Figure 2: **Sensitivity to the BC loss weight.** Online fine-tuning performance across varying BC loss weights ($0.1\times$ to $10\times$ default) without offline data retention. Scaling the BC weight up or down (yellow to dark brown) fails to prevent collapse in both IFQL and FQL. Only removing the BC loss entirely (BC-OFF, green) avoids collapse and enables sustained online learning.

Dissecting IFQL: Isolating the mechanisms of recovery. Our previous experiments reveal that IFQL achieves the most robust performance under the BC-OFF regime. To understand whether this success is driven by IFQL's conservative value updates or its generative policy, we isolate these mechanisms by ablating both (Fig. 4, Col 1). We first ablate the critic by replacing IFQL's IQL-style expectile regression with a standard Bellman double Q-learning loss. Despite this modification, the agent still achieves continuous online improvement when offline data is discarded, and BC is removed. Next, we ablate the actor by replacing the expressive flow-matching model with a standard Gaussian policy. Under this configuration, online performance completely

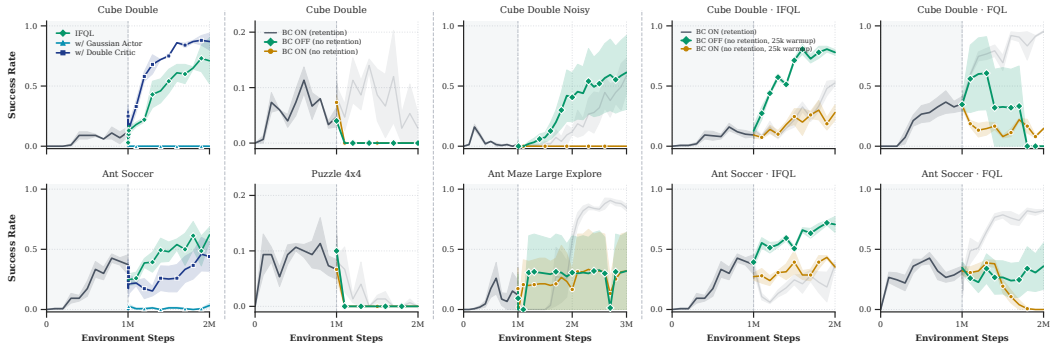

Figure 4: **When does freezing the behavioral prior enable no-retention fine-tuning? Col 1:** In IFQL, fine-tuning succeeds with a standard Bellman double critic (navy) but collapses entirely when its expressive flow prior is ablated in favor of a Gaussian actor (blue). **Col 2:** Removing ReBRAC's BC penalty (BC-OFF) does not prevent collapse (scaled $0.0 - 0.2$ to highlight the immediate drop), demonstrating that omitting BC without a decoupled prior is insufficient. **Col 3:** A frozen flow prior supports continued online improvement without retention even after pretraining on low-quality offline data. **Cols 4 & 5:** Seeding replay with 25k online transitions prevents BC-ON from flatlining across both IFQL and FQL, yet BC-OFF consistently achieves superior asymptotic success.

flatlines. This component isolation demonstrates that IFQL's resilience is not strictly dependent on its specific value-update mechanism but rather on the expressive capacity of its prior.

Unimodal policies cannot be rescued by removing BC. If an expressive prior is truly the bottleneck, then stopping BC should similarly fail to rescue other algorithms restricted to a unimodal representation. To test this, we apply the same no-retention interventions to ReBRAC, an offline RL method utilizing a standard unimodal Gaussian actor. As shown in (Fig. 4, Col 2), discarding the offline dataset causes ReBRAC to completely collapse. Crucially, disabling behavioral cloning (BC-OFF) fails to rescue the policy; performance flat lines at zero alongside BC-ON. This provides further evidence that successfully fine-tuning without offline data may require not only stopping BC but also a sufficiently expressive prior.

Dissecting FQL: Distillation is not an optional regularizer. Under BC-OFF, FQL freezes the flow network but continues distilling its one-step actor toward it. If we sever this link by disabling the distillation loss—forcing the actor to train solely on the value objective—performance catastrophically collapses to zero (Table 5). Unlike IFQL, which samples directly from the flow, FQL relies entirely on distillation to transfer knowledge to the deployed policy. Disconnecting this pathway leaves the actor with no access to the offline manifold. This collapse validates our core design principle: preserving a rich behavioral prior drives online success only when the architecture maintains an active pathway for the decision rule to exploit it.

Can BC-OFF recover performance from low-quality offline data? A potential concern with freezing the behavioral prior is that it permanently anchors the agent to the quality of the offline dataset. We test this by pretraining IFQL on low-quality, suboptimal datasets (`cube-double-noisy` and `antmaze-large-explore`). In these settings, BC-OFF still demonstrates a promising capacity for online policy improvement, achieving returns above the initial offline baseline (Fig. 4, Col 3). Because the flow-based policy is highly expressive, it models the broad, noisy distribution of the offline data without collapsing into a single suboptimal mode. Crucially, it appears to capture enough of the underlying valid action manifold to propose a diverse menu of actions that physically obey the environment dynamics, even if those actions originally led to low reward. We hypothesize that during online fine-tuning, this frozen prior continues to constrain the search space

Table 1: **Earlier BC termination performs best.** Final success after 1M online steps.

τ	`cube-double`	`antsoccer-arena`
0 (immediate)	**0.85**	**0.71**
200k	0.71	0.49
500k	0.53	0.50
800k	0.61	0.51
1M (never)	0.40	0.19

to these physically valid candidates. As the critic improves on new online experiences, it learns to consistently identify and select the higher-value actions from this broad distribution, yielding a better decision rule.

Generalization to Human Teleoperation Demonstrations. To confirm that prior collapse is not an artifact of synthetic OGBench datasets, we evaluate on the Robomimic benchmark using the Multi-Human (`MH`) datasets for `Can` and `Lift`. These datasets comprise 300 demonstrations collected across six operators of varying proficiency, introducing realistic suboptimality and behavioral multimodality. Details can be found in found in section B.3.2. As shown in Figure 5, the same failure mode emerges: continuing online behavioral cloning without offline data retention (BC-ON) induces severe degradation and task collapse across both IFQL and FQL. Conversely, discarding the behavioral objective while retaining the expressive prior (BC-OFF) completely prevents collapse and allows effective finetuning.

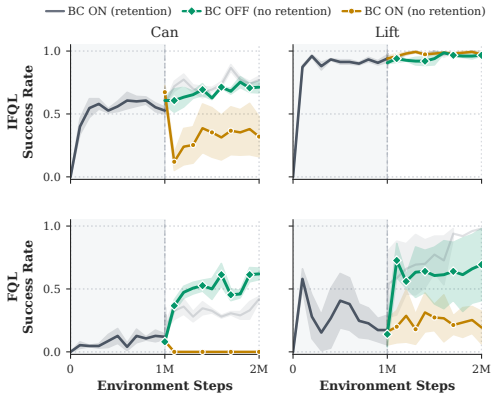

Figure 5: Online fine-tuning on Robomimic under no data retention. Removing the BC objective (BC-OFF) prevents catastrophic degradation and matches offline retention performance across human demonstrations.

Stopping BC late still recovers part of the freezing benefit. Does preserving the behavioral prior require disabling BC from the beginning of online fine-tuning, or can the intervention remain effective after the prior has already been updated on online data? We evaluate intermediate behavioral cloning (BC) termination steps $\tau \in \{200\text{k}, 500\text{k}, 800\text{k}\}$ during 1M online fine-tuning steps to determine if preserving the behavioral prior requires immediate freezing ($\tau = 0$). Table 1 demonstrates that immediate freezing consistently maximizes performance, whereas persistent BC ($\tau = 1\text{M}$) yields the lowest success rates. Terminating BC at any intermediate τ recovers a substantial fraction of this gap. This demonstrates that removing the behavioral objective can act as a robust stabilizing intervention, even after the prior has begun to degrade.

4 DIAGNOSING ONLINE COLLAPSE

The previous section demonstrates that stopping BC can be important for successful no-retention fine-tuning for expressive flow-based policies. But why does continued behavioral cloning fail so catastrophically when offline data is discarded? In this section, we investigate the underlying mechanics of this phenomenon. We trace the failure of BC-ON to a specific behavioral causal chain, moving from catastrophic forgetting of the prior to representational collapse and ultimately critic degeneration. Beyond this feedback loop, we map the degradation of the actor network by tracking progressive layer wise parameter drift. We conclude by ruling out stale optimization statistics to confirm the failure is strictly objective-driven.

Catastrophic forgetting of the behavioral prior. When offline data is discarded, the online replay buffer initially contains only a narrow, correlated slice of recent experience. Continuing to optimize the BC loss on this restricted dataset forces the policy to rapidly overfit to its own early, suboptimal rollouts. This induces catastrophic forgetting of the broad, diverse behavior learned during pretraining. We can directly observe this erasure by tracking the flow policy's action Mean Squared Error (MSE) on a held-out offline dataset (Fig. 6, Left). For both IFQL and FQL, continuing BC without retention causes the offline action MSE to spike immediately and dramatically within the first 100k online steps. The actor rapidly unlearns the valid task manifold. In contrast, removing the behavioral objective (BC-OFF) completely preserves this foundational knowledge. This is also captured by the BC loss on the offline data (Fig. 6, Right).

Online BC induces an actor collapse that drives data drift. This catastrophic forgetting of the prior thoroughly destabilizes the entire offline-to-online transition. To understand the structural impact of this failure, we isolate the actor's intrinsic generative capacity by evaluating it post-hoc on

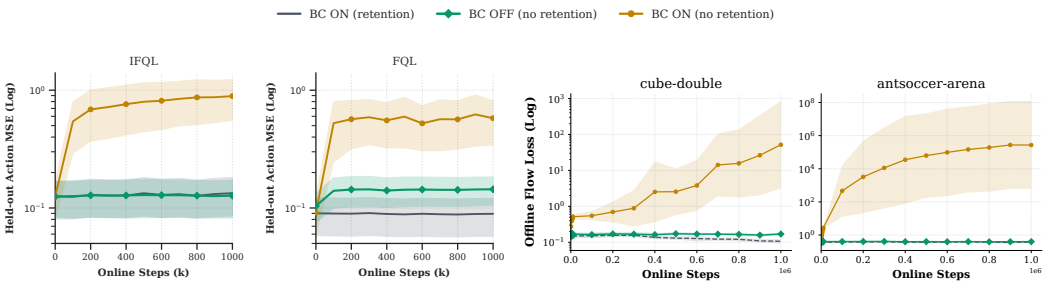

Figure 6: **Continued online-only BC forgets the pretrained behavior. Left and middle:** Held-out offline action MSE for IFQL and FQL, respectively. Continuing BC without retaining offline data sharply increases error, whereas BC-OFF preserves the pretrained behavior. **Right:** BC Loss on the offline data (log scale).

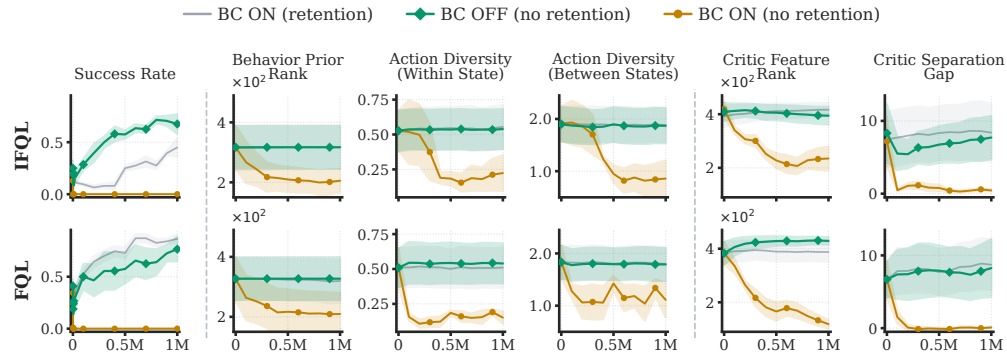

Figure 7: **Online BC induces actor collapse and critic data starvation.** *(*Middle) When BC remains active without offline data, the behavior prior's internal representations degenerate (feature rank collapses), which destroys the policy's cross-state and within-state diversity across both IFQL and FQL. *(Right)* Confined to narrow data collection, the critic is starved of diverse, high-value trajectories. This causes the critic's own feature rank to collapse, destroying its ability to accurately separate high-return states from low-return states (shrinking quantile gap). BC-OFF rescues this entire chain. The details of how these metrics are computed are covered in Appendix C.

a fixed set of diverse, held-out offline validation states. This ensures our measurements reflect the true model capacity rather than artifacts of a collapsed online replay buffer. Our empirical dynamics (Fig. 7, Middle) reveal that as the flow network unlearns the prior, its internal representations rapidly degenerate, causing the rank of its features to plummet. This representational collapse directly destroys the policy's generative capacity, leading to an immediate and sharp drop in both cross-state and within-state action diversity. While cross-state diversity captures whether the policy adapts to different states, within-state diversity ensures the generative model can produce multiple valid actions for a single state. This broad distribution preserves local exploration and, particularly for IFQL, provides the critic with a diverse menu of candidate actions to evaluate. Stripped of this exploratory power, the agent's behavior undergoes a severe spatial collapse. PCA projections of online rollouts confirm this restriction, showing the actor trapped in repetitive failure modes with severely limited state-space coverage (Fig. 8, Left). A concurrent drop in temporal coherency ($\mathbb{E}[\|x_{t+5} - x_t\|]$) for `cube-double` physically captures this collapse, demonstrating that broad exploration degenerates into localized jitter (Fig. 8, Right).

Data starvation breaks the critic. This severely restricted coverage explains the downstream failure of value learning (Fig. 7, Right). The BC objective does not directly alter the critic's loss landscape, and the only mechanism through which BC can impact the critic is through the data the actor collects. Because the actor's spatial exploration has collapsed, it feeds the critic a narrow, repetitive diet of transitions. Starved of diverse data, the critic's own internal representations degenerate, leading to a severe collapse in its feature rank. We believe this representational collapse directly causes the decreasing critic separation gap, the difference in Q-values assigned to state-actions from top-20% versus bottom-20% return trajectories. The critic simply loses its ability to distinguish

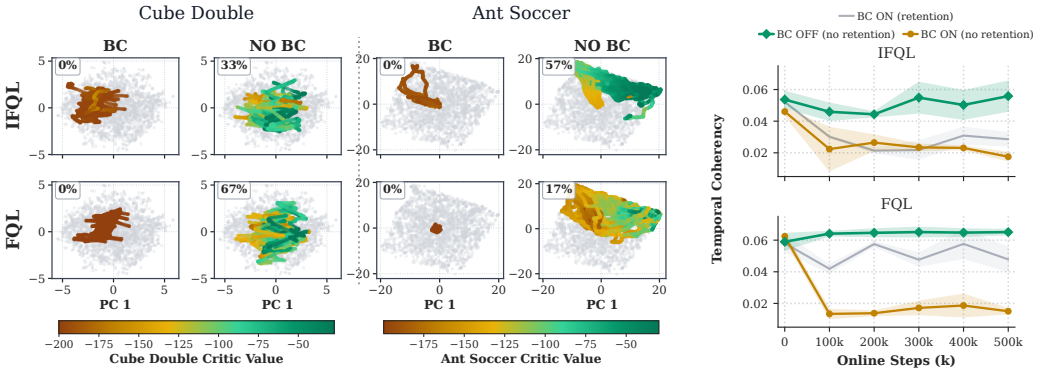

Figure 8: **Spatial collapse and degraded exploration under online Behavioral Cloning. Left:** PCA projections of online rollouts onto the offline manifold at 1.2M steps. Stripped of its exploratory capacity by BC, the agent is forced into repetitive failure modes, leading to severely restricted state-space coverage. **Right:** Short-horizon state displacement on `cube-double`. Continued BC restricts the agent to increasingly localized behavior, while BC-OFF preserves purposeful, wide-area navigation.

good states from bad ones because it no longer sees a sufficiently diverse state-action manifold to maintain rich, discriminative features. BC-OFF prevents this initial diversity collapse, supplying the critic with the rich, stable data necessary to preserve its feature rank and drive continuous policy improvement.

Drift accumulation and non-linear prior degradation. Table 1 shows that intermediate BC termination yields substantial, non-monotonic recovery. Performance drops sharply from the immediate-freeze optimum ($\tau = 0$), plateaus for hundreds of thousands of steps, and collapses under persistent BC ($\tau = 1$M). This trajectory maps directly to the observed parameter drift in (Fig. 9, Left and Middle). The actor's output-adjacent layer diverges within 10^3–10^4 steps, rapidly degrading action quality and permanently lowering the performance ceiling. Deeper representational layers drift more slowly, creating a prolonged window where the prior is damaged but retains sufficient generative capacity to be rescued by removing BC. Persistent BC eventually overwrites these core representations entirely, triggering catastrophic failure.

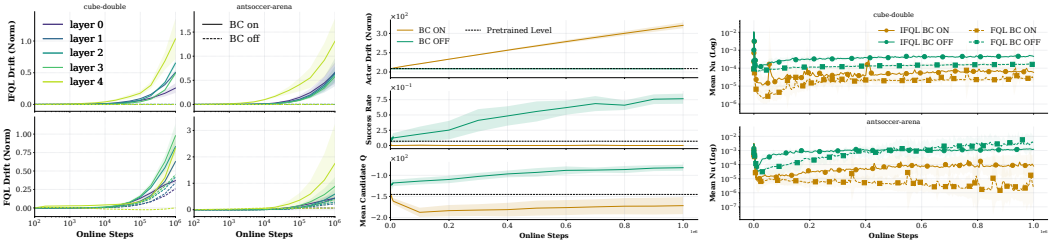

Figure 9: **Online BC drives progressive parameter drift and candidate collapse. Left:** Per-layer relative drift (log x-axis). Under BC-ON (solid), drift penetrates chronologically, beginning with the output-adjacent layer (10^3–10^4 steps). BC-OFF (dashed) exactly freezes IFQL and anchors FQL via distillation. **Middle:** Macroscopic drift and downstream impact for IFQL on `cube-double`. Under BC-ON, unbounded parameter drift occurs from the pretrained initialization. **Right:** Adam's second-moment estimate ($\hat{v}$, log scale). Under BC-ON (solid), gradient magnitude plummets immediately at the onset of online training and remains orders of magnitude below BC-OFF (dashed), indicating a weakened optimization signal.

Ruling out stale-optimizer statistics. The failure of online BC is visible at the level of gradient statistics long before downstream metrics degrade. Under no-retention BC-ON, Adam's second-moment estimate $\hat{v}$ consistently plummets compared to BC-OFF across algorithms and tasks (Fig. 9). Because $\hat{v}$ tracks the running magnitude of squared gradients, this drop indicates that the network is receiving weak, homogeneous gradient signals. Furthermore, because standard offline-to-online RL implementations routinely carry over optimizer statistics from pretraining into the online phase,

this sharp divergence introduces a natural alternative hypothesis: the observed collapse might stem from stale Adam moment estimates failing to adapt to the online distribution shift, rather than the BC objective itself. To isolate this confounder, we clear the optimizer state at the transition boundary. Across all six *Discard*+BC-ON configurations, this reset yields exactly zero performance recovery, with final success rates remaining rigidly at 0.000 (Fig. 10, Left). Under the functional BC-OFF regime, the reset produces only marginal and inconsistent fluctuations (Fig. 10, Right). These results systematically rule out optimization artifacts as a failure mechanism.

5 RELATED WORK

Behavior regularization in offline-to-online RL. Offline RL commonly regularizes policies toward dataset-supported behavior to limit extrapolation error (Fujimoto et al., 2019; Kumar et al., 2019; Levine et al., 2020). Methods such as BCQ, BEAR, BRAC, advantage-weighted regression, TD3+BC, and ReBRAC differ in how they impose this constraint, but all rely on the dataset to define which behaviors should be preserved (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2019; Nair et al., 2020; Kostrikov et al., 2021; Fujimoto & Gu, 2021; Tarasov et al., 2023). Most offline-to-online methods retain the offline dataset during fine-tuning, whereas considerably less work studies the setting in which that supervision disappears. WSRL discards the dataset but stabilizes the transition through pretrained-policy rollouts, while PORL considers a policy-only initialization with a newly learned critic (Zhou et al., 2025; Xiao et al., 2025). We retain the offline-pretrained actor and critic and study behavior regularization after its defining data become unavailable. Our results show that behavioral fitting should stop once that supervision disappears. See App. A for a broader discussion.

Expressive policies for data-driven RL. Diffusion-QL combines an expressive diffusion actor with denoising-based BC and value maximization (Wang et al., 2023). IDQL instead models the behavior distribution with diffusion and obtains an improved decision rule by critic-guided reweighting or selection (Hansen-Estruch et al., 2023). FQL uses conditional flow matching to learn a behavior model and distills value-improved actions into an efficient one-step actor (Park et al., 2025). Q-chunking extends expressive behavior modeling to temporally extended action sequences (Li et al., 2025). Related work on policy-agnostic improvement also separates value improvement from the policy class used to represent the resulting actions (Mark et al., 2024). Recent methods such as EXPO and FINO further investigate stable online optimization with expressive policies (Dong et al., 2026c; Shin et al., 2026). We study a distinct question: whether an expressive behavioral model can remain useful as a fixed prior after both its training data and its behavioral-fitting objective are removed.

6 LESSONS AND OPPORTUNITIES

Lessons. No-retention fine-tuning is fundamentally distinct from conventional offline-to-online RL. When the offline dataset is discarded, behavioral cloning ceases to act as a stabilizing regularizer; instead, it forces the network to imitate its own narrowing online experience. As shown, this rapidly cannibalizes the behavioral prior, causing a representational collapse that starves the rest of the learning process. The solution is not simply to turn off the objective, but to embrace a two-part design principle: successfully transitioning to online-only data requires (1) freezing an inherently expressive, multimodal behavioral prior, and (2) maintaining a decoupled decision mechanism (such as critic-guided selection or distillation) that allows online value learning to exploit that frozen manifold.

Opportunities. Our findings open several new avenues for data-restricted RL. While a strictly frozen prior is highly effective when pretraining and fine-tuning share the same underlying dynamics, extreme environmental distribution shifts may eventually require the prior to adapt. This motivates the need for mechanisms that bridge the gap between catastrophic online-only BC and hard freezing. Future work could explore generative replay, uncertainty-aware plasticity, or selective behavioral updates that preserve the broad, valid modes of the offline dataset while safely incorporating novel online behaviors. This is especially relevant to VLA fine-tuning, where expressive pretrained policies may need to support online improvement after demonstration data are discarded (Dong et al., 2026a;b). Ultimately, treating the pretrained model as a permanent, expressive foundation rather than a strictly plastic network offers a robust path forward for RL deployments with strict data constraints.

AI USE STATEMENT

In this work, we used generative AI tools for aiding and polishing paragraph-level writing, discussing and refining the structure and flow of the introduction, finding related work, and editing and writing the code. We also used generative AI tools to assist with early drafts of some sections, which were later substantially rewritten and replaced by the authors. We have not used generative AI tools for research ideation, and the rest of the required disclosure tasks, such as generating synthetic datasets or proving mathematical claims, are not applicable to this work. We have reviewed the AI-assisted work. We reviewed LLM-generated code and read and incorporated suggested related work papers manually. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Reincarnating reinforcement learning: Reusing prior computation to accelerate progress. *Advances in neural information processing systems*, 35:28955–28971, 2022.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pp. 1577–1594. PMLR, 2023.

Perry Dong, Kuo-Han Hung, Tian Gao, Dorsa Sadigh, and Chelsea Finn. Expo-ft: Sample-efficient reinforcement learning finetuning for vision-language-action models, 2026a. URL https://arxiv.org/abs/2605.25477.

Perry Dong, Kuo-Han Hung, Dorsa Sadigh, and Chelsea Finn. Reinforcement learning for real-time vision-language-action policies, 2026b. URL https://arxiv.org/abs/2609.18207.

Perry Dong, Qiyang Li, Dorsa Sadigh, and Chelsea Finn. Expo: Stable reinforcement learning with expressive policies. In *International Conference on Learning Representations*, volume 2026, pp. 131379–131394, 2026c.

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

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

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, 2023. URL https://arxiv.org/abs/2304.10573.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning, 2021. URL https://arxiv.org/abs/2110.06169.

Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. *Stabilizing off-policy Q-learning via bootstrapping error reduction*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit Under-Parameterization Inhibits Data-Efficient Deep Reinforcement Learning, October 2021. URL http://arxiv.org/abs/2010.14498. arXiv:2010.14498 [cs].

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. URL https://arxiv.org/abs/2005.01643.

Jianxiong Li, Xiao Hu, Haoran Xu, Jingjing Liu, Xianyuan Zhan, and Ya-Qin Zhang. PROTO: Iterative policy regularizied offline-to-online reinforcement learning, 2024. URL https://openreview.net/forum?id=S77skzM12O.

Qiyang Li and Sergey Levine. Q-learning with adjoint matching, 2026. URL https://arxiv.org/abs/2601.14234.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=XUks1Y96NR.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

Max Sobol Mark, Tian Gao, Georgia Gabriela Sampaio, Mohan Kumar Srirama, Archit Sharma, Chelsea Finn, and Aviral Kumar. Policy agnostic rl: Offline rl and online rl fine-tuning of any class and backbone. *arXiv preprint arXiv:2412.06685*, 2024.

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

Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=GcEIvidYSw.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KVf2SFL1pi.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2021. URL https://openreview.net/forum?id=ToWi1RjuEr8.

Yongjae Shin, Jongseong Chae, Jongeui Park, and Youngchul Sung. Flow matching with injected noise for offline-to-online reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=6wd38R8L0Z.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AHvFDPi-FA.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning, 2019. URL https://arxiv.org/abs/1911.11361.

Wei Xiao, Jiacheng Liu, Zifeng Zhuang, Runze Suo, Shangke Lyu, and Donglin Wang. Efficient online rl fine tuning with offline pre-trained policy only, 2025. URL https://arxiv.org/abs/2505.16856.

Haichao Zhang, Wei Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-Y34L45JR6z.

Yi Zhao, Rinu Boney, Alexander Ilin, Juho Kannala, and Joni Pajarinen. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning, 2022. URL https://arxiv.org/abs/2210.13846.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HN0CYZbAPw.

APPENDIX CONTENTS

A Expanded Related Work **14**

B Experiment Details **15**

B.1 Computational Resources 15

B.2 Implementation and Hyperparameters 15

B.3 Domain Details 17

C Mechanistic Analysis Metric Definitions **17**

D Experiments **18**

A Expanded Related Work

Offline reinforcement learning is vulnerable to extrapolation error because value functions and policies may be evaluated on actions that are poorly represented in the dataset (Fujimoto et al., 2019; Kumar et al., 2019; Levine et al., 2020). A broad family of methods therefore constrains policy improvement toward the offline behavior distribution. BCQ restricts action selection using a learned behavior model (Fujimoto et al., 2019); BEAR and BRAC penalize divergence from the behavior policy (Kumar et al., 2019; Wu et al., 2019); and advantage-weighted methods regress toward dataset actions according to their estimated value (Peng et al., 2021; Nair et al., 2020; Kostrikov et al., 2021). TD3+BC and ReBRAC combine value maximization with direct regression toward actions in the replay buffer (Fujimoto & Gu, 2021; Tarasov et al., 2023). Although these methods differ in form, they share an important assumption: the data defining the behavioral constraint remain available during optimization. The behavioral objective is meaningful because its supervision continues to represent the desired action distribution. We study what happens when the objective remains active but the data that originally defined it do not.

Offline-to-online reinforcement learning initializes an online learner from offline pretraining and subsequently improves it through environment interaction. AWAC uses offline data to initialize a policy and then performs advantage-weighted online updates (Nair et al., 2020). Other methods retain offline transitions through balanced replay, large update-to-data ratios, or conservative value estimation (Lee et al., 2022; Ball et al., 2023; Nakamoto et al., 2023). Cal-QL explicitly calibrates the offline critic so that its values provide a more suitable initialization for online fine-tuning (Nakamoto et al., 2023). Additional methods manage the stability–plasticity trade-off using separate offline and online policies, adaptive behavioral constraints, or evolving reference policies (Zhang et al., 2023; Zhao et al., 2022; Li et al., 2024). These approaches show that offline supervision can accelerate online learning, but they generally assume continued access to the offline dataset or introduce an additional mechanism for preserving its influence.

A smaller body of work studies online adaptation after offline data are discarded. WSRL shows that offline data need not be retained when the online replay buffer is initialized with rollouts from the pretrained policy (Zhou et al., 2025). Its analysis emphasizes critic recalibration and value divergence under the distribution shift from offline data to online experience. PORL considers an even more restrictive policy-only setting in which the pretrained policy is available but the offline critic is not, and a new critic must be learned from interaction (Xiao et al., 2025). Reincarnating RL studies the broader problem of reusing previously learned policies and value functions to accelerate subsequent reinforcement learning rather than restarting from scratch (Agarwal et al., 2022). These works establish that no-retention fine-tuning is possible, but primarily focus on preserving or reconstructing value estimates. We retain the offline-pretrained actor and critic and study a complementary actor-side question: should behavioral fitting continue once the data that defined its objective become unavailable?

Another line of work treats offline-to-online fine-tuning as a replay composition problem. Balanced replay and related methods control the relative contribution of offline and online transitions through sampling ratios, prioritization, or adaptive buffer construction (Lee et al., 2022; Ball et al., 2023). These methods primarily change the data seen by the critic and actor. Our intervention is different: we change whether the behavioral model continues to optimize an objective whose supervision has changed. This distinction is important in the no-retention setting. Once the offline buffer is unavailable, replay manipulation alone cannot restore the original behavioral distribution unless that distribution has been preserved by the model itself.

Diffusion- and flow-based policies provide expressive alternatives to unimodal Gaussian actors and can represent multimodal conditional action distributions (Wang et al., 2023; Liu et al., 2022; Lipman et al., 2023). Diffusion-QL combines behavior modeling with value maximization (Wang et al., 2023), while IDQL uses a diffusion behavior model to propose actions and a critic to reweight or select among those proposals (Hansen-Estruch et al., 2023). FQL replaces iterative diffusion with flow matching and distills value-improved actions into an efficient one-step actor (Park et al., 2025). These methods separate the distribution used to propose actions from the mechanism used to improve or select them.

Recent work also studies how expressive policies should be optimized online. EXPO avoids directly propagating unstable value gradients through a long-horizon generative policy and instead uses a

lightweight edit policy to modify actions sampled from an expressive base policy (Dong et al., 2026c). FINO adapts flow policies online through noise injection and entropy-guided sampling to promote exploration beyond the offline distribution (Shin et al., 2026). These methods continue adapting the expressive policy while attempting to stabilize its optimization. We study a different intervention: preserving the pretrained expressive prior and continuing online improvement through the critic and decision rule.

Action chunking extends expressive behavior modeling from individual actions to temporally extended action sequences. Q-chunking applies this idea directly to reinforcement learning, using chunked actions to preserve temporally coherent behavior, improve exploration, and enable multi-step value backups (Li et al., 2025). This setting is especially relevant to no-retention fine-tuning because updating a chunk prior on a small online buffer can erase not only individual actions but also longer-horizon temporal structure. We therefore evaluate whether the prior-preservation principle extends from single-action flow policies to action-chunked RL.

Our work differs from these studies in its central intervention and diagnostic question. We do not introduce a new expressive policy class, a new critic-calibration procedure, or a new replay rule. Instead, we isolate the consequences of continuing versus stopping behavioral-model fitting after the offline data disappear. The resulting comparison reveals when an expressive prior remains useful, when removing BC is insufficient, and why the connection between the prior and the deployed decision rule is essential.

B Experiment Details

B.1 Computational Resources

To optimize computational efficiency and ensure strictly matched starting conditions, we decoupled the offline pretraining phase from the online fine-tuning phase. For each algorithm, task, and random seed, we first executed the 1M offline gradient steps and saved the resulting pretrained checkpoints. All subsequent online fine-tuning setups were then initialized from these shared checkpoints.

Experiments were executed on NVIDIA H100 GPUs. To maximize hardware utilization, we multiplexed six individual experimental runs per GPU, with the underlying allocation providing 12 CPU cores and 124 GB of system memory under a 12-hour wall-clock budget. In total, the full study includes several hundred individual online training runs across the four evaluated algorithms (FQL, IFQL, ReBRAC, and QC).

B.2 Implementation and Hyperparameters

We base our implementations of FQL, IFQL, and ReBRAC on the official FQL repository, and Action-Chunked RL (QC) on the official QC repository. To ensure fair evaluations, we do not re-tune the base optimization hyperparameters. Instead, we retain the standard architecture and optimization settings from the reference implementations and apply their recommended task-specific hyperparameters. These values remain strictly fixed across all experiments so that our ablations isolate only the specific architectural or algorithmic components under study.

For FQL, the actor, flow model, and critic use four-layer MLPs with hidden dimensions $(512, 512, 512, 512)$ and GELU activations. Optimization uses Adam with a learning rate of 3×10^{-4}, minibatches of size 256, and a target-network smoothing coefficient $\tau = 0.005$. Flow policies use linear flow-matching paths, uniformly sampled flow times, and 10 Euler integration steps. FQL uses two Q-functions; their mean is used in the actor objective and, for the OGBench tasks considered here, also for the critic target.

IFQL uses the same flow-matching configuration as FQL and follows the default IQL value-learning setup, utilizing an IQL expectile of $\kappa = 0.9$. At evaluation time, IFQL samples N candidate actions from the behavioral flow prior and executes the action with the highest estimated value. We use the task-specific values of N detailed in Table 3.

For QC, we preserve the flow-matching setup and maintain a fixed action chunking horizon of $H = 5$ across all tasks. Similar to IFQL, QC evaluates by sampling N candidate action sequences from the frozen prior and selects the sequence that maximizes the chunk-level critic.

ReBRAC uses identical default architecture and optimization settings unless otherwise specified. Its primary method-specific hyperparameters are the actor behavioral-cloning coefficient α_1 and the critic behavioral-cloning coefficient α_2.

Table 2: Base architecture and optimization hyperparameters.

Hyperparameter	**Value**
Default MLP hidden dimensions	$(512, 512, 512, 512)$
Activation	GELU
Optimizer	Adam
Learning rate	3×10^{-4}
Batch size	256
Target-network update rate τ	0.005
Discount γ	0.99 default; 0.995 for humanoidmaze/antsoccer
Number of Q-functions	2
FQL / IFQL / QC flow policy	
Flow time sampling distribution	$\mathrm{Unif}([0, 1])$
Euler integration steps	10
IFQL	
IQL expectile κ	0.9
QC	
Action chunking horizon H	5

Table 3: Task-specific hyperparameters adopted from the FQL and QC papers. IFQL and QC report the number of candidate actions or sequences N, FQL reports its behavioral-regularization coefficient α, and ReBRAC reports the actor and critic BC coefficients (α_1, α_2).

Task	**IFQL** N	**FQL** α	**QC** N	**ReBRAC** (α_1, α_2)
`cube-double`	32	300	32	$(0.1, 0)$
`puzzle-4x4`	32	1000	32	$(0.3, 0.01)$
`antsoccer-arena`	64	30	64	$(0.01, 0.01)$
`humanoidmaze-medium`	32	100	32	$(0.01, 0.01)$
`humanoidmaze-large`	32	100	32	$(0.01, 0.01)$

Table 4: Training and evaluation protocol used in our experiments.

Setting	**Value**
Offline pretraining steps	1,000,000
Online fine-tuning steps	1,000,000
Replay buffer capacity	2,000,000 transitions
Evaluation episodes per checkpoint	50
Evaluation interval	Every 100,000 steps
Seeds per configuration	3

B.3 Domain Details

B.3.1 OGBench

We cover both manipulation and locomotion domains from the OGBench benchmark using the single-task variants of each environment. To construct our evaluation suite, we systematically select three specific tasks per domain: the environment's designated default task, supplemented by the first two available tasks in numerical order. For example, because the default task for `antsoccer-arena` is task 4, our evaluation spans tasks 1, 2, and 4. Under the single-task variant, the reward structure is semi-sparse: locomotion tasks receive a binary 0 or 1 reward upon completion, while manipulation tasks receive a negative reward corresponding to the number of unaccomplished subtasks.

- `cube-double`: A manipulation task to control a robot arm to arrange two cubes into designated configurations within a 500-step episode. Tasks 1, 2, 3.
- `puzzle-4x4`: A manipulation task requiring the agent to press physical buttons to solve a 4×4 "Lights Out" puzzle within a 500-step episode. Tasks 1, 2, 4.
- `antsoccer-arena`: A locomotion task where an Ant agent must dribble a soccer ball to a target location in an open arena within a 1000-step episode. Tasks 1, 2, 4.
- `humanoidmaze-medium`: A locomotion task requiring a 21-DoF Humanoid agent to navigate to a goal location in a medium-sized maze within a 2000-step episode. Tasks 1, 2, 3.
- `humanoidmaze-large`: A locomotion task requiring the 21-DoF Humanoid agent to navigate to a goal location in a larger and more complicated maze layout within a 2000-step episode. Tasks 1, 2, 3.

Table 5: **FQL requires a connection to the frozen flow.** Final success under no-retention fine-tuning. Freezing the behavioral flow succeeds only when the one-step actor remains distilled toward it.

Task	BC-On	BC-Off	BC-Off, no distillation
`cube-double`	0.000	$\mathbf{0.977 \pm 0.014}$	0.000
`antsoccer-arena`	0.000	0.303 ± 0.206	0.000

B.3.2 Robomimic

To evaluate whether the collapse of the behavioral prior generalizes beyond synthetic benchmarks, we evaluate on the Robomimic benchmark (Mandlekar et al., 2021) using human-collected teleoperation datasets. We adopt the Multi-Human (`MH`) datasets collected via RoboTurk, where demonstrations were gathered across six human operators stratified into three distinct proficiency levels (experienced, adequate, and inexperienced; 50 demonstrations per operator). Each task dataset contains 300 successful demonstrations featuring natural variance, suboptimal paths, and multimodal execution strategies. We evaluate on the following tasks:

- **`Lift`:** Requires a 7-DoF Franka Emika Panda arm to lift a small cube above a height threshold.
- **`Can`:** Requires the robot arm to grasp a coke can and place it inside a designated container bin.

C Mechanistic Analysis Metric Definitions

In section 4, we track several metrics to quantify the representational and behavioral collapse of the actor and critic. To decouple the intrinsic capacity of the networks from the evolving quality of the online replay buffer, these metrics are evaluated post-hoc on a fixed, held-out offline validation set.

Actor Feature Rank. This metric measures the effective dimensionality of the actor network's penultimate representation. We pass 1,024 offline validation states through the actor and extract the penultimate layer activations (dimension 512) to form a feature matrix X. We center the features ($X_{\text{centered}} = X - \mu_X$) and compute the singular values s_i using SVD. We compute the threshold stable rank (Kumar et al., 2021) as the minimum number of singular values required to capture 99% of the total singular value energy: $\sum_{i=1}^{k} s_i / \sum_i s_i \geq 0.99$. A higher rank indicates that the policy maintains rich feature representations without collapsing into a low-dimensional subspace.

Action Diversity (Within State). This metric measures the proposal stochasticity and sensitivity of the actor to input noise at a given state. For 128 validation states, we sample multiple action candidates per state using different random noise seeds. We compute the standard deviation of the actions at each state and report the mean across all states. High within-state diversity indicates the generative model can produce multiple distinct valid actions; low values indicate a deterministic collapse.

Action Diversity (Between States). This metric measures the spatial dispersion of predicted actions across different states. We predict the mean action $a_i = \pi(s_i)$ for $K = 128$ distinct validation states and compute the mean pairwise Euclidean distance across all state pairs: $\frac{1}{K(K-1)} \sum_{i \neq j} \|a_i - a_j\|_2$. If this metric collapses near zero, the actor is outputting a constant action regardless of the input state.

Critic Feature Rank. This metric measures the effective representation rank of the Q-function network. We pass 1,024 offline (s, a) pairs through the critic, extract the penultimate layer activations, and compute the threshold stable rank exactly as done for the actor (with a 99% cumulative energy threshold). This measures whether the critic maintains expressive, high-capacity features capable of accurate value prediction.

Critic Separation Gap This metric quantifies the ability of the critic to separate high-quality transitions from poor ones. We partition the offline validation dataset by trajectory return, isolating the top 20% of transitions ($(s_{\text{top}}, a_{\text{top}})$) and the bottom 20% ($(s_{\text{bot}}, a_{\text{bot}})$). We evaluate the ensemble minimum Q-values for both subsets and compute the difference: $\mathbb{E}[Q(s_{\text{top}}, a_{\text{top}})] - \mathbb{E}[Q(s_{\text{bot}}, a_{\text{bot}})]$. A large positive gap indicates the critic correctly assigns higher values to high-return states, whereas a gap collapsing to zero indicates severe value distortion.

D EXPERIMENTS

Optimizer-state reset. One possible explanation for the offline-to-online collapse is that the optimizer carries stale Adam statistics from offline pretraining into a different online data distribution. If this were the primary cause, resetting Adam's first- and second-moment estimates at the offline-to-online transition should restore performance, particularly in the failing DISCARD+BC-ON condition.

We therefore repeat the corresponding experiments with the optimizer state reset immediately before online fine-tuning, while keeping the pretrained network parameters and all other hyperparameters unchanged. As shown in Figure 10, resetting Adam does not recover performance in any of the six DISCARD+BC-ON cells: all remain at zero change. Under BC-OFF, the reset produces only small and inconsistent differences. These results indicate that stale optimizer statistics do not explain the collapse; the dominant failure is associated with continuing behavioral fitting on online-only data.

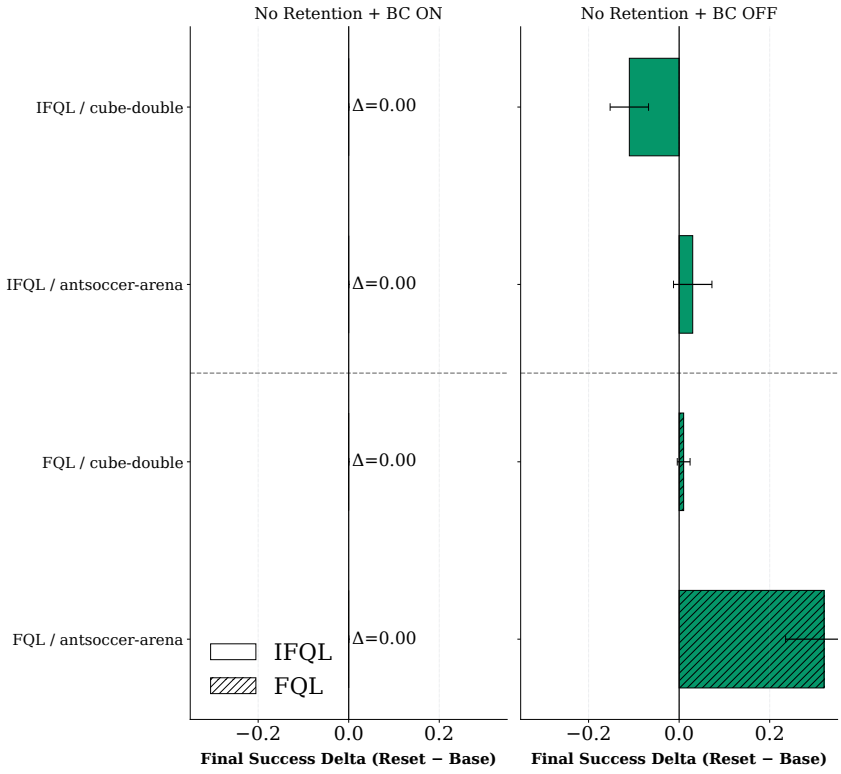

Figure 10: **Resetting the optimizer's Adam state** at the online switch never rescues the discard+BC-on collapse (left, all six cells at exactly zero change) and has only small, inconsistent effects once BC is already off (right). The failure mode is not stale optimizer statistics.