BELLMAN-AWARE CONSERVATIVE INTERVENTION: RECURSIVE ACTOR–FLOW SELECTION FOR OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-based offline reinforcement learning can combine an expressive flow policy with an efficient one-step actor, yielding two candidate actions at each state. Selecting between them is not purely a local ranking problem because the value of a current action can depend on future actor–flow selections. We introduce Bellman-Aware Conservative Intervention (BACI), an offline actor-flow selection algorithm that keeps both action generators fixed and learns a dedicated decision critic. Its Bellman target recursively accounts for future actor–flow selections. At deployment, a flow proposal can replace the actor action when its estimated advantage exceeds a replacement cost, with accepted replacements executed with probability p. Under explicit assumptions, we establish exact policy improvement within the restricted actor–flow policy class, identify conditions under which recursive selection captures gains missed by actor-policy evaluation, and characterize the effect of value-ranking errors under approximate deployment. Experiments with frozen actor–flow pairs trained by Guided Flow Policy (GFP) and Flow Actor-Critic (FAC) show average improvements over the corresponding pretrained actors on OGBench and D4RL. Fitted Q Evaluation (FQE) and original-critic controls show that recursive value learning provides additional benefits beyond direct ranking, with its advantage over FQE varying across benchmarks. code at anonymous.4open.science/r/ICLR2027-BACI

1 INTRODUCTION

Offline reinforcement learning (RL) learns a policy from a fixed dataset without additional environment interaction (Levine et al., 2020; Fu et al., 2020). This setting avoids costly or unsafe online exploration but is sensitive to distribution shift and inaccurate values for poorly supported actions. Recent diffusion and flow policies provide expressive alternatives to standard unimodal policies and can represent complex behavior distributions (Wang et al., 2023; Lipman et al., 2023; Park et al., 2025b). Flow-based offline RL methods often pair such a flow policy with an efficient one-step actor.

Existing methods improve how the policy or value function is learned from the offline data. Behavior regularization constrains policy updates toward the data, conservative methods suppress unreliable values for unsupported actions, and in-sample methods avoid querying out-of-distribution actions during value learning (Fujimoto et al., 2019; Kumar et al., 2019; 2020; Fujimoto & Gu, 2021; Kostrikov et al., 2022; Wu et al., 2019; Tarasov et al., 2023). Within flow-based offline RL, Flow Q-Learning (FQL) combines a behavior flow with a one-step actor, Guided Flow Policy (GFP) makes flow imitation value-aware, and Flow Actor-Critic (FAC) uses the learned flow distribution to regularize critic learning (Park et al., 2025b; Tiofack et al., 2026; Chae et al., 2026). These methods do not explicitly learn the value of repeatedly selecting between the actor and flow outputs. Their critics evaluate actions assuming future actions follow the policy used in the Bellman backup, whereas deployment with repeated actor–flow selection can produce a different sequence of actions. The relative value of the two candidates can therefore depend on selections made at later states, motivating a value function defined for repeated actor-flow selection.

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.

We introduce Bellman-Aware Conservative Intervention (BACI), which learns this selection policy from offline transitions while keeping the pretrained actor and flow policies fixed. The actor provides the default action and the flow policy an alternative proposal. A separate decision critic uses a Bellman target that recursively includes future actor–flow selections. At deployment, a flow proposal must satisfy the distance constraint and exceed the actor by the replacement cost before it can be selected; accepted proposals are executed with probability p. Under explicit assumptions, we establish policy improvement for exact optimization within the restricted actor–flow policy class and give a sufficient condition under which recursive evaluation captures gains missed by actor-policy evaluation. We also analyze the effect of ranking errors for an approximate decision critic.

Experiments use frozen actor–flow pairs trained by GFP and FAC. For each pair, only the BACI decision critic is learned. Across OGBench and D4RL, BACI improves average performance over the corresponding pretrained actor on the evaluated task sets. Fitted Q Evaluation (FQE) (Le et al., 2019; Zhang et al., 2022; Wang et al., 2024) and direct selection with the original critic isolate the contribution of recursive value learning. The matched results show additional gains over FQE on OGBench, while the relative advantage is more task dependent on D4RL. Component and deployment-probability analyses are provided in the appendix.

2 Preliminaries and Problem Setup

2.1 Offline RL and Actor–Flow Policies

We consider a discounted Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \mu_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $P(\cdot \mid s, a)$ is the transition kernel, $r(s, a)$ is the reward, μ_0 is the initial-state distribution, and $\gamma \in (0, 1)$ is the discount factor. We use a normalized continuous action space $\mathcal{A} = [-1, 1]^{d_a}$, where d_a is the action dimension.

The agent is trained from a fixed offline dataset

$$\mathcal{D} = \{(s_i, a_i, r_i, s'_i, m_i)\}_{i=1}^{N},$$

where (s_i, a_i, r_i, s'_i) is a recorded transition and $m_i \in \{0, 1\}$ indicates whether Bellman continuation is active after that transition. No additional environment interaction is used during training.

For a policy π, we define

$$J(\pi) = \mathbb{E}_{\substack{s_0 \sim \mu_0 \\ a_k \sim \pi(\cdot|s_k)}} \left[\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) \right],$$

and denote its state- and action-value functions by V^π and Q^π, respectively. They satisfy

$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[Q^\pi(s', a') \right]. \tag{1}$$

We also use the normalized discounted state-visitation distribution

$$d^\pi = (1 - \gamma) \sum_{k=0}^{\infty} \gamma^k \, \mathrm{Law}_\pi(s_k).$$

Recent flow-based offline RL methods combine a one-step actor with a flow policy (Park et al., 2025b; Tiofack et al., 2026; Chae et al., 2026). For state s, the actor produces

$$a_A = \mathrm{clip}\big(A_\theta(s, z_A), -1, 1\big),$$

where z_A denotes the actor's random input when the actor is stochastic. The flow policy starts from an independent sample z_F and generates an action by numerically integrating the learned velocity field v_ω. With M Euler steps,

$$x_0 = z_F, \qquad x_{j+1} = x_j + \frac{1}{M} v_\omega \left(s, x_j, \frac{j}{M} \right), \qquad a_F = \mathrm{clip}(x_M, -1, 1).$$

Thus, the actor produces a_A with a single network evaluation, whereas the flow policy produces a_F through iterative integration. Both policies are pretrained and remain fixed during BACI training.

2.2 Repeated Actor–Flow Selection

At each state, the actor provides the default action a_A, while the flow policy provides an alternative proposal a_F. We study how to select between these two actions when the same choice can arise again at later states. A value function trained for the actor policy evaluates a current action assuming that subsequent actions are generated by the actor. This assumption no longer matches the deployed policy when later states can again select between the actor and flow proposal. The value of a current candidate can therefore depend on future actor–flow selections, rather than only on the action taken at the current state. Our goal is to learn this repeated selection policy from the offline dataset while keeping both action generators fixed. BACI addresses this problem by learning a separate decision critic whose Bellman target accounts for future actor–flow selections, as described in Section 3.

3 BACI: Learning Actor–Flow Selection

BACI learns to select between the pretrained actor and flow policy defined in Section 2. Both action generators remain fixed. The additional decision critic is trained from offline transitions to account for actor–flow selections at future states.

3.1 Candidate Generation and Selection Constraints

At each state s, independent samples $z_A, z_F \sim \mathcal{N}(0, I_{d_a})$ generate one actor action and one flow proposal:

$$\begin{aligned} a_A &= \mathrm{clip}(A_\theta(s, z_A), -1, 1), \\ x_0 &= z_F, \qquad x_{\ell+1} = x_\ell + \frac{1}{M} v_\omega(s, x_\ell, \ell/M), \quad \ell = 0, \dots, M-1, \\ a_F &= \mathrm{clip}(x_M, -1, 1). \end{aligned} \tag{2}$$

Here I_{d_a} is the identity matrix, M is the integration-step count used by the pretrained flow policy, and clipping is applied coordinatewise. These action-generation procedures follow the corresponding pretrained models (Park et al., 2025b; Tiofack et al., 2026; Chae et al., 2026). The actor uses one network evaluation, while the flow uses M velocity-field evaluations.

Let $K = (z_A, z_F)$ denote the candidate randomness and let $\pi_A(\cdot \mid s) = \mathrm{Law}(a_A(s, K))$ be the actor policy. The proposed replacement is restricted by

$$\begin{aligned} d(s, K) &= \frac{\|a_F - a_A\|_2^2}{4 d_a}, \\ e(s, K) &= \mathbf{1}\{d(s, K) \le r_{\max},\ a_A \neq a_F\}, \\ c(s, K) &= c_0 + \lambda_d d(s, K). \end{aligned} \tag{3}$$

The eligibility indicator e is additionally set to zero if either candidate contains a non-finite entry. For valid clipped actions, $d \in [0, 1]$. The replacement cost c is expressed on the same scale as the action values. The main experiments use $(r_{\max}, c_0, \lambda_d) = (0.15, 0.10, 0.25)$.

The actor remains available without a replacement cost. The distance constraint limits the displacement of an accepted action, while the cost requires a larger estimated gain for more distant proposals. Neither quantity estimates the probability that an action is supported by the offline dataset.

3.2 Recursive Value Learning

A value function for the actor alone evaluates an action assuming that subsequent actions follow π_A. Repeated actor–flow selection requires a different value function because the flow proposal can also be selected at later states.

Let $t \in [0, 1]$ be the probability that an eligible flow proposal is available for selection during value learning. Conditional on this opportunity, a selection rule may choose either candidate. Otherwise, it must choose the actor. Denote the resulting class of stationary selection rules by Π_t. A rule in Π_t specifies its decision after observing the sampled candidates, so it also specifies whether a replacement cost is incurred.

For a selection policy π, define

$$C(\pi) = \mathbb{E}_\pi \sum_{k=0}^{\infty} \gamma^k c(s_k, K_k) H_k, \qquad J_c(\pi) = J(\pi) - C(\pi),$$

where $H_k \in \{0, 1\}$ indicates an executed flow replacement at environment step k. The objective J_c deducts each executed replacement cost once; the recorded environment reward is unchanged.

Assumption 1 (Selection and evaluation conditions)**.** *The state space is a standard Borel space and the task transition kernel is Markov. Rewards satisfy $|r(s, u)| \le R_{\max}$, and replacement costs satisfy $0 \le c(s, K) \le c_{\max}$. The fixed candidate maps and eligibility rule are measurable and produce finite actions almost surely. At each decision, candidate randomness is drawn independently of the past conditional on the current state. The opportunity variable is independent of the candidate randomness and the past, with the stated probability. Ties retain the actor action. The analysis uses bounded measurable value functions and $\gamma \in (0, 1)$. Bellman termination is represented by an absorbing state with zero subsequent reward, zero replacement cost, and zero value.*

For a bounded scalar action-value function q, define the cost-adjusted difference

$$\Delta_q(s, K) = q(s, a_F) - q(s, a_A) - c(s, K).$$

The optimized state value and its Bellman operator are

$$\begin{aligned} \mathcal{V}_t[q](s) &= \mathbb{E}_K[(1 - te)q(s, a_A) + te \max\{q(s, a_A), q(s, a_F) - c\}] \\ &= \mathbb{E}_K[q(s, a_A) + te[\Delta_q(s, K)]_+], \\ (\mathcal{T}_t q)(s, u) &= r(s, u) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,u)} \mathcal{V}_t[q](s'), \end{aligned} \tag{4}$$

where $[x]_+ = \max(x, 0)$ and $u \in \mathcal{A}$ denotes the action being evaluated. The arguments (s, K) of e and c are omitted inside the expectation.

The action value in Equation (4) includes the current environment reward and future cost-adjusted values. The current replacement cost is deducted when choosing between the two actions in $\mathcal{V}_t$, not a second time in $\mathcal{T}_t$. Setting $t = 0$ recovers evaluation of π_A. Setting $t = 1$ permits every eligible proposal to be considered, but still allows the actor to be selected.

The following result explains why retaining the actor as a zero-cost option supports policy improvement.

Proposition 1 (Exact actor-relative improvement)**.** *Under Assumption 1, $\mathcal{T}_t$ is a γ-contraction in the supremum norm and has a unique fixed point q_t^*. The selection rule induced by q_t^*, using opportunity probability t, attains the largest value of J_c over Π_t. Denote this policy by π_t^*. Then*

$$J_c(\pi_t^*) \ge J(\pi_A), \qquad J(\pi_t^*) \ge J(\pi_A) + C(\pi_t^*). \tag{5}$$

The optimal cost-adjusted state value is nondecreasing in t.

The proof is in Appendix B.1. The comparison holds because always choosing the actor is feasible for every t. Strict improvement is possible but is not asserted in every task. The Bellman construction uses standard fitted Q-iteration and dynamic programming with stochastic action availability (Ernst et al., 2005; Boutilier et al., 2018).

Exact actor-policy evaluation followed by a cost-adjusted selection step can also improve the actor. Recursion becomes useful when an initial flow action requires another flow selection to produce a gain. The following construction makes this distinction explicit.

Proposition 2 (Gains requiring repeated selection)**.** *Consider two nonterminal states x, y, deterministic candidates, and initial state x. At both states, the actor terminates with reward zero. The flow action at x yields reward $-\kappa$, with $\kappa > 0$, and transitions to y. At y, it yields reward R and terminates. Both flow actions are eligible, with costs $c_x, c_y \ge 0$. For $p \in (0, 1]$, suppose*

$$R > c_y, \qquad \kappa + c_x < \gamma p(R - c_y).$$

A selector based on exact Q^{π_A} rejects the flow action at x and obtains zero return. The selector based on q_1^* accepts both flow actions. Executing each accepted replacement with independent probability p gives

$$J_c(\widehat{\pi}_p) = p[-\kappa - c_x + \gamma p(R - c_y)] > 0 = J(\pi_A).$$

Appendix B.2 proves the result. The proposition identifies a setting in which repeated selection provides a strict gain over exact one-step actor-policy evaluation. It does not require the flow-only policy to be better in general.

Algorithm 1 BACI training and action selection. The pretrained action generators remain fixed; only the additional decision critics are optimized.

Require: Offline data $\mathcal{D}$; pretrained actor A_θ, flow v_ω, and target critic parameters $\bar{\phi}_1, \bar{\phi}_2$; integration count M; selection parameters $(r_{\max}, c_0, \lambda_d, t, p)$.

1: Set $\psi_j = \bar{\psi}_j = \bar{\phi}_j$ and initialize new Adam states for $j = 1, 2$.
2: **for** $n = 1, \ldots, U$ **do**
3: Sample a minibatch of B transitions $(s_i, a_i, r_i, s_i', m_i)$ from $\mathcal{D}$.
4: Generate one independent actor–flow pair at each s_i' using Equation (2).
5: Compute d_i', e_i', c_i' using Equation (3).
6: Form the stopped targets in Equation (6).
7: Update ψ_1, ψ_2 by Adam on $\mathcal{L}_Q$.
8: Update $\bar{\psi}_j \leftarrow (1-\tau)\bar{\psi}_j + \tau\psi_j$.
9: **end for**
10: Freeze the fitted decision critics.
11: At each deployment state, generate a fresh candidate pair and apply Equation (7).

3.3 Decision-Critic Training

The implementation approximates Equation (4) with two decision critics q_{ψ_1}, q_{ψ_2} and target parameters $\bar{\psi}_1, \bar{\psi}_2$. Both sets of parameters are initialized from the pretrained target critics $Q_{\bar{\phi}_1}, Q_{\bar{\phi}_2}$:

$$\psi_j = \bar{\psi}_j = \bar{\phi}_j, \qquad j \in \{1, 2\}.$$

The decision critics use new Adam optimizer states. The pretrained actor, flow policy, and original critic parameters are not updated.

For each logged transition $(s_i, a_i, r_i, s_i', m_i)$ in minibatch $\mathcal{B}$, sample one actor action and one flow proposal at s_i' using Equation (2). Define $\bar{q}(s,u) = \min_{j\in\{1,2\}} q_{\bar{\psi}_j}(s,u)$. The shared regression target and loss are

$$\begin{aligned} y_i &= r_i + \gamma m_i \Big[\bar{q}(s_i', a_{A,i}') + t e_i' \big[\bar{q}(s_i', a_{F,i}') - \bar{q}(s_i', a_{A,i}') - c_i' \big]_+ \Big], \\ \mathcal{L}_Q(\psi) &= \frac{1}{2|\mathcal{B}|} \sum_{i\in\mathcal{B}} \sum_{j=1}^{2} \big[q_{\psi_j}(s_i, a_i) - \mathrm{sg}(y_i) \big]^2 . \end{aligned} \tag{6}$$

Primes indicate quantities evaluated at the next state, and sg stops gradients through the target. The minimum over critic heads is computed separately for each candidate. The loss is fitted at the logged action a_i. No reward or transition is assigned to an unobserved candidate.

The opportunity probability t is included analytically in the target; training transitions are not discarded by sampling an opportunity coin. After each Adam update, the target parameters are updated using

$$\bar{\psi}_j \leftarrow (1-\tau)\bar{\psi}_j + \tau\psi_j.$$

The experiments use $U = 50{,}000$ updates, batch size $B = 256$, learning rate 3×10^{-4}, and $\tau = 0.005$. Twin critics, target averaging, and Adam follow standard practice (Fujimoto et al., 2018; Kingma & Ba, 2015). Their use does not provide a statistical bound on decision-critic error.

3.4 Policy Extraction

At deployment, the fitted online critics provide the score $\widehat{q}(s,u) = \min_{j\in\{1,2\}} q_{\psi_j}(s,u)$. Let

$$\widehat{I}(s,K) = \mathbf{1}\{\widehat{q}(s,a_F) - \widehat{q}(s,a_A) > c(s,K)\}.$$

With an independent $Z \sim \mathrm{Bernoulli}(p)$, return

$$a_{\mathrm{out}}(s,K,Z) = \begin{cases} a_F, & Ze(s,K)\widehat{I}(s,K) = 1, \\ a_A, & \text{otherwise.} \end{cases} \tag{7}$$

The induced action distribution is $\widehat{\pi}_p(\cdot \mid s) = \mathrm{Law}(a_{\mathrm{out}}(s, K, Z))$. Rejection returns the actor action generated in the same call. There is no second actor draw, action averaging, or additional proposal search.

The main setting is $t = 1$ and $p = 0.5$. Training therefore evaluates future selections with full opportunity, while deployment executes each accepted replacement with probability p. This setting is fixed across the reported tasks. The matched Fitted Q Evaluation (FQE) variant uses $t = 0$ with the same deployment rule (Le et al., 2019); the $p = 1$ variant reuses the recursively trained critic without probabilistic rejection.

Because t and p differ in the main setting, Proposition 1 is not by itself a guarantee for the implemented policy. To analyze this policy, let

$$\Delta^*(s, K) = q_1^*(s, a_F) - q_1^*(s, a_A) - c(s, K), \qquad I^* = \mathbf{1}\{\Delta^* > 0\}.$$

The next result separates the effect of not executing a useful flow action from the effect of an incorrect value comparison.

Theorem 1 (Loss from skipped selections and ranking errors)**.** *Under Assumption 1, for any bounded measurable fitted score $\widehat{q}$ and $p \in [0, 1]$,*

$$J_c(\pi_1^*) - J_c(\widehat{\pi}_p) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\widehat{\pi}_p}, K}\left[e\left((1-p)[\Delta^*]_+ + p|\Delta^*|\mathbf{1}\{\widehat{I} \neq I^*\}\right)\right]. \tag{8}$$

Define the candidate-difference error

$$E_{\mathrm{pair}} = [\widehat{q}(s, a_F) - \widehat{q}(s, a_A)] - [q_1^*(s, a_F) - q_1^*(s, a_A)].$$

Whenever $\widehat{I} \neq I^*$, $|\Delta^*| \leq |E_{\mathrm{pair}}|$.

The proof is in Appendix B.3. Equation (8) relates performance directly to errors that change the selection decision, rather than to all absolute value errors. An additive error shared by both candidate values at a state cancels from E_{pair}. Reducing p also increases the probability of skipping useful proposals, and changes $d^{\widehat{\pi}_p}$. The theorem therefore does not imply that a smaller p always improves performance.

A direct comparison with the actor gives a positive sufficient condition for probabilistic execution. Fix $\widehat{q}$ and define

$$\begin{aligned} h(s, K) &= e(s, K)\widehat{I}(s, K), \\ G(s, K) &= Q^{\pi_A}(s, a_F) - Q^{\pi_A}(s, a_A), \\ g(s) &= \mathbb{E}_K[h(s, K)(G(s, K) - c(s, K))], \\ A &= \mathbb{E}_{s\sim d^{\pi_A}} g(s), \qquad \bar{h} = \mathbb{E}_{s\sim d^{\pi_A}, K} h(s, K), \\ S_g &= \sup_s g(s) - \inf_s g(s). \end{aligned}$$

Here A is the average net local gain under the actor's state distribution, $\bar{h}$ is the corresponding acceptance probability, and S_g is the range of the net local gain.

Theorem 2 (Actor-relative improvement under probabilistic execution)**.** *Under Assumption 1, with the fitted score and candidate generators held fixed,*

$$J_c(\widehat{\pi}_p) - J(\pi_A) \geq \frac{pA}{1-\gamma} - \frac{\gamma p^2 \bar{h} S_g}{(1-\gamma)^2}. \tag{9}$$

If $A > 0$ and $\bar{h}S_g > 0$, strict improvement holds for

$$0 < p \leq 1, \qquad p < \frac{A(1-\gamma)}{\gamma \bar{h} S_g}.$$

If $A > 0$ and $\bar{h}S_g = 0$, every $p \in (0, 1]$ gives strict improvement in J_c, and hence also in J.

Appendix B.5 proves the result. The bound compares a local gain that is linear in p with a state-distribution term that is quadratic in p. It gives a sufficient condition for beneficial replacements to remain useful after they change the visited states. However, A involves the true actor value and is not determined by the fitted acceptance test alone. The bound does not select $p = 0.5$ or certify it from the observed TD loss. Appendix B.4 gives a complementary condition in terms of actor-relative action-value error.

Table 1: OGBench success rate (%). Published baselines (†) are taken from GFP Table 2 for IQL, ReBRAC, FQL, and GFP, and from FAC Table 1 for FAC (Tiofack et al., 2026; Chae et al., 2026). The two BACI columns apply the same BACI algorithm to the pretrained actor–flow pair produced by GFP or FAC, respectively; the corresponding actor and flow policy are kept fixed while the BACI decision critic is trained. For each BACI training seed, every goal is evaluated over 100 episodes. The five goal scores are averaged to obtain the category score, and we report the mean ± sample standard deviation of these category scores over 10 independent training seeds. Published baselines retain the statistics and evaluation protocols of their original sources. "-" indicates a task setting not reported in the corresponding GFP or FAC source paper; the corresponding BACI experiment is therefore not included. Bold indicates the largest displayed mean in each row; ties are all bolded.

	Gaussian Policies †		Flow-based Methods †			BACI (ours)	
Category	IQL	ReBRAC	FQL	GFP	FAC	GFP actor–flow	FAC actor–flow
antmaze-large-navigate	53 ± 3	95.9 ± 0.4	88.1 ± 3.4	93.8 ± 1.5	92.6 ± 2.5	**97.6** ± 0.8	93.6 ± 1.3
antmaze-large-stitch	30.4 ± 3.2	**89.2** ± 6.6	58.1 ± 8.7	68.9 ± 0.8	-	76.7 ± 0.5	-
antmaze-large-explore	12.9 ± 1.7	82.7 ± 7.6	87.9 ± 6.6	91.9 ± 0.9	-	**92.6** ± 0.4	-
antmaze-giant-navigate	4 ± 1	33.2 ± 5.7	16.3 ± 8.2	27.9 ± 8.5	23.0 ± 5.1	28.8 ± 2.6	**35.2** ± 2.8
humanoid-medium-navigate	33 ± 2	59.2 ± 12.1	58 ± 5	72.0 ± 2.8	75.6 ± 3.6	**78.4** ± 1.7	77.2 ± 1.9
humanoid-medium-stitch	27.3 ± 2.9	61.1 ± 8.2	63.2 ± 6.7	66.2 ± 5.7	-	**73.1** ± 3.3	-
humanoid-large-navigate	2 ± 1	12.9 ± 4.2	6.5 ± 2.7	17.8 ± 9.6	8.3 ± 5.5	**34.4** ± 4.9	7.6 ± 2.5
antsoccer-navigate	8 ± 2	55.9 ± 1.5	60 ± 4	57.9 ± 1.9	67.7 ± 2.9	**70.4** ± 2.8	69.0 ± 1.6
antsoccer-stitch	2.8 ± 1	22.0 ± 1.5	28.6 ± 2.3	30.5 ± 2.2	-	**37.3** ± 3.6	-
cube-single-play	83 ± 3	91 ± 2	96 ± 1	98.8 ± 0.4	98.8 ± 1.4	**99.4** ± 0.6	98.6 ± 1.2
cube-single-noisy	53.2 ± 4.1	98.4 ± 0.6	**100** ± 0	**100** ± 0	-	**100** ± 0.0	-
cube-double-play	7 ± 1	12.6 ± 1.8	29 ± 2	47.2 ± 1.6	33.1 ± 5.7	**48.0** ± 3.1	37.2 ± 2.5
cube-double-noisy	4.5 ± 0.8	19.6 ± 2.1	38.2 ± 5.3	63.1 ± 3.3	-	**67.2** ± 1.5	-
cube-triple-play	0.1 ± 0.1	2.9 ± 1.2	3.9 ± 1.5	15.9 ± 2.0	-	**18.5** ± 2.2	-
cube-triple-noisy	4.8 ± 1.2	5.2 ± 2.9	3.5 ± 1.6	**24.5** ± 2.8	-	24.1 ± 3.9	-
puzzle-3×3-play	9 ± 1	21 ± 1	30 ± 1	23.1 ± 2.2	**100.0** ± 0.0	30.2 ± 1.8	**100.0** ± 0.0
puzzle-4×4-play	7 ± 1	17.1 ± 1.3	17 ± 2	26.1 ± 2.1	32.3 ± 6.5	29.6 ± 2.2	**32.8** ± 2.4
puzzle-4×4-noisy	0.1 ± 0.0	1.1 ± 0.3	15.6 ± 1.1	18.8 ± 1.7	-	**21.3** ± 2.3	-
scene-play	28 ± 1	41.6 ± 3.6	56 ± 2	53.5 ± 2.9	71.3 ± 7.6	54.4 ± 1.6	**74.3** ± 3.4
scene-noisy	16.0 ± 1.2	39.9 ± 2.6	59.3 ± 1.4	57.5 ± 0.9	-	**62.6** ± 0.5	-
visual manipulation	42 ± 4	60 ± 2	65 ± 2	62.8 ± 1.5	76.0 ± 3.9	72.3 ± 3.8	**79.5** ± 2.2

4 EXPERIMENTS

Experiments on OGBench and D4RL evaluate the performance of BACI and the contribution of recursive actor–flow value learning. The same selection algorithm is applied to actor–flow pairs trained by Guided Flow Policy (GFP) and Flow Actor-Critic (FAC) (Tiofack et al., 2026; Chae et al., 2026). Comparisons with published methods assess benchmark performance, while matched experiments examine the effects of the decision critic and the action-selection rule.

Experimental setup. For each pretrained pair, the actor and flow policy remain fixed while the additional decision critic is trained for 50K updates using Equation (6). The main results use training opportunity probability $t = 1$ and deployment probability $p = 0.5$ throughout. Training recipes, hyperparameters, and complete task specifications are provided in Appendix C.4.

Table 1 reports OGBench success rates (Park et al., 2025a) across the categories covered by the corresponding source methods, including the visual manipulation setting. For each training seed, every goal is evaluated over 100 episodes. The five goal scores are averaged within each category, and the table reports the mean and sample standard deviation of these category scores over 10 independent training seeds. Table 2 reports normalized returns on D4RL (Fu et al., 2020), also over 10 training seeds. Each seed is evaluated over 50 episodes on AntMaze and 10 episodes on MuJoCo and Adroit.

IQL, ReBRAC, FQL, GFP, and FAC results retain the training and evaluation protocols of their cited sources. We therefore use them as benchmark-level comparisons rather than as matched estimates of the isolated BACI effect. Additional matched actor–flow controls using fixed pretrained pairs are reported in Appendix D.

OGBench performance. Table 1 shows competitive performance across navigation and manipulation tasks. Across the two reported actor–flow inputs, at least one BACI configuration attains or ties the largest displayed mean in 19 of the 21 categories. With GFP-trained inputs, BACI achieves

Table 2: D4RL normalized return. Published baselines (†) are taken from FAC Table 2, with GFP results and Adroit-expert comparisons taken from GFP Table 11 (Tiofack et al., 2026; Chae et al., 2026). The two BACI columns apply the same BACI algorithm to pretrained actor–flow pairs produced by GFP or FAC, respectively; the corresponding actor and flow policy remain fixed while the BACI decision critic is trained. All BACI results are reported as mean ± sample standard deviation over 10 independent training seeds. For each training seed, AntMaze tasks are evaluated over 50 episodes, while MuJoCo and Adroit tasks are evaluated over 10 episodes. Published baselines retain the statistics and evaluation protocols of their original sources. "-" indicates a task setting not reported in the corresponding GFP or FAC source paper; the corresponding BACI experiment is therefore not included. Bold indicates the largest displayed mean in each row. Abbreviations m, mr, and me denote medium, medium-replay, and medium-expert, respectively. No cross-paper average is reported because task coverage and evaluation protocols differ across columns.

	Gaussian Policies †		Flow-based Methods †			BACI (ours)	
Task	IQL	ReBRAC	FQL	GFP	FAC	GFP actor–flow	FAC actor–flow
MuJoCo							
halfcheetah-m	47.4	**65.6**	60.3 ± 1.1	-	65.0 ± 1.5	-	64.1 ± 1.8
hopper-m	66.3	**102.0**	68.1 ± 3.4	-	91.9 ± 3.9	-	98.9 ± 2.3
walker2d-m	78.3	82.5	77.2 ± 2.5	-	85.2 ± 0.9	-	**85.5** ± 0.6
halfcheetah-mr	44.2	51.0	49.3 ± 0.5	-	55.4 ± 2.7	-	**56.4** ± 1.1
hopper-mr	94.7	98.1	49.8 ± 7.2	-	99.1 ± 0.9	-	**99.3** ± 0.5
walker2d-mr	73.9	77.3	53.1 ± 7.9	-	83.0 ± 5.8	-	**87.7** ± 2.9
halfcheetah-me	86.7	101.1	99.6 ± 6.5	-	101.9 ± 5.6	-	**102.1** ± 3.7
hopper-me	91.5	**107.0**	83.1 ± 17.0	-	104.2 ± 4.6	-	106.7 ± 2.2
walker2d-me	109.6	**111.6**	106.1 ± 1.8	-	108.4 ± 0.6	-	110.2 ± 0.4
AntMaze							
antmaze-umaze	87.5	97.8	96.0	96.8 ± 1.9	98.5 ± 3.0	98.2 ± 1.2	**99.4** ± 1.8
antmaze-umaze-diverse	62.2	88.3	89.0	91.9 ± 2.7	93.5 ± 6.0	92.6 ± 1.5	**96.2** ± 2.3
antmaze-m-play	71.2	84.0	78.0	81.9 ± 5.2	88.0 ± 9.6	86.4 ± 2.4	**88.8** ± 3.6
antmaze-m-diverse	70.0	76.3	71.0	61.6 ± 20.9	85.0 ± 7.3	73.8 ± 6.1	**87.4** ± 4.7
antmaze-large-play	39.6	60.4	84.0	82.6 ± 5.4	90.0 ± 4.3	84.0 ± 2.9	**91.2** ± 2.5
antmaze-large-diverse	47.5	54.4	83.0	84.1 ± 5.4	88.0 ± 6.0	86.8 ± 2.6	**89.6** ± 3.8
Adroit							
pen-human	81.5	**103.5**	53.0	64.6 ± 5.4	73.9 ± 14.7	71.9 ± 3.6	78.7 ± 4.9
pen-cloned	77.2	91.8	74.0	77.1 ± 10.4	103.2 ± 11.1	87.8 ± 4.5	**108.5** ± 2.4
pen-expert	133.6	**154.1**	142	140.4 ± 4.7	-	148.4 ± 3.2	-
door-human	3.1	0.0	0.0	0.3 ± 0.3	5.5 ± 3.3	0.8 ± 0.2	**6.2** ± 2.7
door-cloned	0.8	1.1	2.0	1.6 ± 1.9	4.1 ± 3.9	2.1 ± 1.3	**9.1** ± 2.2
door-expert	**105.3**	104.6	104	104.1 ± 0.6	-	105.1 ± 1.3	-
hammer-human	2.5	0.2	1.0	4.4 ± 4.9	8.6 ± 5.4	5.2 ± 2.6	**9.5** ± 3.8
hammer-cloned	1.1	6.7	11.0	12.4 ± 5.4	11.1 ± 11.2	**13.6** ± 3.7	11.7 ± 6.6
hammer-expert	129.6	**133.8**	125	123.6 ± 2.0	-	128.5 ± 1.6	-
relocate-human	0.1	0.0	0.0	0.5 ± 0.3	0.6 ± 0.5	0.8 ± 0.4	**0.9** ± 0.3
relocate-cloned	0.2	0.9	0.0	**1.6** ± 0.7	0.5 ± 0.4	1.5 ± 0.7	1.1 ± 0.5
relocate-expert	106.5	106.6	107	103.2 ± 3.7	-	**108.2** ± 4.5	-

the highest means on AntMaze-large-navigate, both Humanoid navigation categories, AntSoccer-navigate, and several Cube tasks. For example, its score on Humanoid-large-navigate is 34.4, compared with 17.8 for the published GFP result. With FAC-trained inputs, BACI achieves the highest means on AntMaze-giant-navigate, Puzzle-4×4-play, Scene-play, and visual manipulation.

The results also retain the differences between the two input pairs and across tasks. ReBRAC has the highest score on AntMaze-large-stitch, while GFP is slightly higher on Cube-triple-noisy. On Cube-single-noisy and Puzzle-3×3-play, the best BACI results match baselines that already reach the maximum success rate. The table therefore shows gains across several settings rather than a uniform advantage for every task and input pair.

D4RL performance. Table 2 covers MuJoCo locomotion, AntMaze navigation, and Adroit manipulation. At least one of the two BACI configurations has the largest displayed mean on 18 of the 27 tasks. With FAC-trained inputs, BACI obtains the highest means on all six AntMaze tasks and five of the nine MuJoCo tasks. On Adroit, the strongest displayed results include pen-cloned and door-cloned with FAC-trained inputs, and hammer-cloned and relocate-expert with GFP-trained inputs.

The gains differ across datasets. ReBRAC remains strongest on pen-human, pen-expert, and hammer-expert, while IQL has the highest mean on door-expert. These results place BACI alongside strong Gaussian- and flow-based methods across different data regimes. Because the published results use their original protocols, the following matched comparisons examine the effect of adding BACI to a fixed pretrained pair.

Actor–flow selection analyses. Appendix D reports the complete measured actor–flow selection controls, including actor-only and flow-only execution, direct selection with the original critic, Fitted Q Evaluation (FQE) (Le et al., 2019), and BACI. Appendix D.4 further examines recursive value learning and deployment probability on the Puzzle tasks. The matched results show that the effect of recursive selection and execution probability varies across tasks and pretrained actor–flow pairs. Additional reward and replacement-cost diagnostics are reported in Appendix D.3.

5 Conclusion

We introduced Bellman-Aware Conservative Intervention (BACI), an offline policy-improvement method for pretrained actor–flow policies. BACI keeps both action generators fixed and learns a decision critic whose Bellman target accounts for future actor–flow selections, allowing current actions to be evaluated under repeated selection rather than actor-only future actions. Under explicit assumptions, we established policy improvement for exact recursive selection and characterized how value-ranking errors affect approximate deployment. Experiments with actor–flow pairs trained by GFP and FAC show that the same BACI mechanism can improve the corresponding pretrained actors across OGBench and D4RL. Matched comparisons with FQE and direct selection using the original critic further show that recursive value learning can provide gains beyond additional critic fitting or one-step action ranking, although the magnitude of this benefit varies across benchmarks. Overall, BACI provides a simple way to improve existing actor–flow policies by learning how their actions should be selected over time, without retraining the underlying action generators.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and manuscript polishing, and to support literature search and discovery of relevant prior work. All retrieved references were independently checked by the authors, who reviewed the final manuscript and take responsibility for its content.

REPRODUCIBILITY STATEMENT

Algorithm 1 and Appendix C specify the refinement and deployment procedures. Appendix B provides full proofs under stated assumptions. Appendix C.4 defines evaluation groups and reporting rules, including the separation of imported baseline results from new measurements. The accompanying materials retain goal-level result tables, source-protocol attribution, and the scripts used to aggregate the current measurements. Native policies, decision critics, and deployment-only diagnostics are identified separately so that imported evaluation records are not counted as additional runs.

ETHICS STATEMENT

This work studies policy learning from existing simulation datasets and does not establish safety for physical deployment. A distance test or a positive estimated value margin should not be interpreted as a real-world safety certificate. Deployment in settings involving people or physical systems requires application-specific validation beyond the experiments considered here.

REFERENCES

Craig Boutilier, Alon Cohen, Avinatan Hassidim, Yishay Mansour, Ofer Meshi, Martin Mladenov, and Dale Schuurmans. Planning and learning with stochastic action sets. In *International Joint Conference on Artificial Intelligence*, 2018. URL https://www.ijcai.org/proceedings/2018/650.

David Brandfonbrener, Will Whitney, Rajesh Ranganath, and Joan Bruna. Offline rl without off-policy evaluation. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/274a10ffa06e434f2a94df765cac6bf4-Abstract.html.

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. URL https://arxiv.org/abs/2602.18015.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005. URL https://www.jmlr.org/papers/v6/ernst05a.html.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. URL https://arxiv.org/abs/2004.07219.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2106.06860.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023. URL https://arxiv.org/abs/2304.10573.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.11239.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/jin21e.html.

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.20081.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c2073ffa77b5357a498057413bb09d3a-Abstract.html.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.04779.

Romain Laroche, Paul Trichelair, and Rémi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/laroche19a.html.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/le19a.html.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Natinael Solomon Neggatu, Jeremie Houssineau, and Giovanni Montana. Evaluation-time policy switching for offline reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, 2025. URL https://www.ifaamas.org/Proceedings/aamas2025/pdfs/p1520.pdf.

Thanh Nguyen and Chang D. Yoo. One-step flow q-learning: Addressing the diffusion policy bottleneck in offline reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2508.13904.

Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. Is value learning really the main bottleneck in offline rl? In *Advances in Neural Information Processing Systems*, volume 37, pp. 79029–79056, 2024.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ecd92623ac899357312aaa8915853699-Abstract-Conference.html.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *International Conference on Machine Learning*, 2025b. URL https://proceedings.mlr.press/v267/park25f.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2011.13456.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.09836.

Franki Nguimatsia Tiofack, Théotime Le Hellard, Fabian Schramm, Nicolas Perrin-Gilbert, and Justin Carpentier. Guided flow policy: Learning from high-value actions in offline reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2512.03973.

Jiayi Wang, Zhengling Qi, and Raymond K. W. Wong. A fine-grained analysis of fitted q-evaluation: Beyond parametric models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

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*, 2023. URL https://openreview.net/forum?id=AHvFDPi-FA.

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

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2106.06926.

Ruiqi Zhang, Xuezhou Zhang, Chengzhuo Ni, and Mengdi Wang. Off-policy fitted q-evaluation with differentiable function approximators: Z-estimation and inference theory. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Zhiyuan Zhou, Andy Peng, Charles Xu, Qiyang Li, Tobias Springenberg, Kevin Frans, and Sergey Levine. Test-time gradient guidance of flow policies in reinforcement learning. *arXiv preprint arXiv:2606.11087*, 2026. URL https://arxiv.org/abs/2606.11087.

A Related Work

Flow-based policies for offline reinforcement learning. Expressive generative policies have been introduced to offline RL to represent complex and multimodal action distributions beyond standard Gaussian policies (Ho et al., 2020; Song et al., 2021; Wang et al., 2023; Hansen-Estruch et al., 2023; Kang et al., 2023; Nguyen & Yoo, 2026). Flow Q-Learning (FQL) combines a behavior flow with a one-step actor, using the flow to regularize actor optimization without backpropagating through iterative action generation (Park et al., 2025b). Guided Flow Policy (GFP) further makes flow imitation value-aware and couples the actor and flow through mutual guidance (Tiofack et al., 2026). Flow Actor-Critic (FAC) uses a flow behavior proxy to regularize critic learning and reduce unreliable values for out-of-distribution actions (Chae et al., 2026). These methods change how the actor, flow policy, or critic is trained. BACI considers a subsequent policy-improvement problem: given a pretrained actor and flow policy, it learns a decision critic for repeated actor–flow selection while leaving both action generators fixed.

Test-time policy improvement and policy switching. Evaluation-Time Policy Switching dynamically switches between separately trained offline RL and behavior-cloning policies using epistemic uncertainty from a critic ensemble together with a dataset-derived measure of aleatoric uncertainty (Neggatu et al., 2025). Q-Guided Flow (QGF) improves a pretrained flow policy at test time by using critic gradients to modify the flow sampling process (Zhou et al., 2026). BACI also performs policy improvement after the actor and flow policies have been trained, but follows a different mechanism. It does not modify either candidate or the flow sampling trajectory. Instead, one actor action and one flow proposal are generated first, and a separately trained decision critic selects between them. The decision critic is trained with a Bellman target that accounts for future actor–flow selections, rather than ranking the current candidates in isolation.

Offline policy evaluation and policy improvement. Fitted Q Evaluation (FQE) estimates the value function of a fixed policy from offline data (Le et al., 2019), and recent work continues to study its statistical properties under flexible function approximation (Zhang et al., 2022; Wang et al., 2024). Policy evaluation can be followed by a constrained improvement step, and even a single policy-improvement step can provide a strong offline RL baseline (Brandfonbrener et al., 2021). Recent empirical studies further show that policy extraction and test-time policy improvement can be as important as value learning itself (Park et al., 2024). These observations are directly relevant to BACI. Our matched FQE baseline evaluates the actor policy and then applies the same actor–flow deployment rule, whereas the BACI decision critic evaluates actions while accounting for future actor–flow selections. Baseline-relative policy improvement has also been studied through methods such as Safe Policy Improvement with Baseline Bootstrapping (SPIBB), which restricts policy changes in state–action regions with insufficient data support (Laroche et al., 2019). BACI does not estimate this type of statistical uncertainty or interpret actor–flow distance as a measure of dataset support. Its analysis instead considers improvement within the fixed actor–flow policy class under explicit assumptions on value-estimation error.

B Proofs and Additional Characterizations

All results concern the fixed actor and flow generators and the selection rules defined in Section 3. The initial-state distribution is μ_0. For compatibility with compact operator notation, write $b(s, K) = a_A(s, K)$, $a(s, K) = a_F(s, K)$, and $\pi_b = \pi_A$. These symbols do not introduce additional policies or candidates.

A selection policy includes its decision conditional on the sampled candidates. Its induced action distribution determines environment transitions, while its executed-replacement indicator determines the replacement cost. All value functions are extended by zero at the absorbing terminal state.

B.1 Proof of Proposition 1

Proof. For a fixed state and candidate pair, abbreviate $q_A = q(s, a_A)$ and $q_F = q(s, a_F)$. For two bounded measurable functions q, q',

$$\begin{aligned}&|\max\{q_A, q_F - c\} - \max\{q'_A, q'_F - c\}| \\ &\quad \le \max\{|q_A - q'_A|,\ |q_F - q'_F|\} \le \|q - q'\|_\infty.\end{aligned}$$

The coefficients $1 - te$ and te are nonnegative and sum to one. Taking expectations gives

$$\|\mathcal{V}_t[q] - \mathcal{V}_t[q']\|_\infty \le \|q - q'\|_\infty.$$

Since a Markov expectation is nonexpansive in the supremum norm,

$$\|\mathcal{T}_t q - \mathcal{T}_t q'\|_\infty \le \gamma \|q - q'\|_\infty.$$

Bounded measurable functions, extended by zero at the terminal state, form a complete space under this norm. Assumption 1 ensures that $\mathcal{T}_t$ maps this space into itself. Banach's fixed-point theorem therefore gives a unique fixed point q_t^*.

To establish optimality, define for a bounded state value V,

$$q_V(s, u) = r(s, u) + \gamma \mathbb{E}_{s'|s,u} V(s'),$$

and

$$(\mathcal{B}_t V)(s) = \mathbb{E}_K \left[q_V(s, a_A) + te[q_V(s, a_F) - q_V(s, a_A) - c]_+ \right].$$

The operator $\mathcal{B}_t$ is monotone and a γ-contraction. It maximizes the current environment reward minus the current replacement cost plus the discounted next-state value over the permitted choices.

Let V_t^* be its fixed point. Then

$$q_t^* = r + \gamma P V_t^*, \qquad V_t^* = \mathcal{V}_t[q_t^*].$$

A maximizing selection rule is measurable because it compares two measurable scores and uses a fixed tie-break. For the maximizing policy π_t^*, its cost-adjusted evaluation operator satisfies

$$\mathcal{B}_c^{\pi_t^*} V_t^* = \mathcal{B}_t V_t^* = V_t^*.$$

Uniqueness of policy evaluation yields $V_c^{\pi_t^*} = V_t^*$.

For any other $\pi \in \Pi_t$,

$$\mathcal{B}_c^{\pi} V_t^* \le \mathcal{B}_t V_t^* = V_t^*.$$

Monotonicity and repeated policy evaluation give

$$V_c^{\pi} = \lim_{n \to \infty} (\mathcal{B}_c^{\pi})^n V_t^* \le V_t^*.$$

Thus π_t^* maximizes J_c over Π_t.

Always selecting the actor is feasible for every t and incurs zero replacement cost. Hence

$$V_t^* \ge V^{\pi_A}, \qquad J_c(\pi_t^*) \ge J(\pi_A).$$

Because replacement costs are nonnegative,

$$J(\pi_t^*) = J_c(\pi_t^*) + C(\pi_t^*) \ge J(\pi_A) + C(\pi_t^*).$$

This proves Equation (5).

Finally, if $0 \le t_1 \le t_2 \le 1$, then for every bounded V,

$$(\mathcal{B}_{t_2} V)(s) - (\mathcal{B}_{t_1} V)(s) = (t_2 - t_1) \mathbb{E}_K e[q_V(s, a_F) - q_V(s, a_A) - c]_+ \ge 0.$$

Starting from $V_{t_1}^*$ and iterating $\mathcal{B}_{t_2}$ produces a nondecreasing sequence converging to $V_{t_2}^*$. Therefore $V_{t_2}^* \ge V_{t_1}^*$. This ordering concerns exact optimal cost-adjusted values. □

B.2 Actor Evaluation and Repeated Selections

Proof of Proposition 2.

Proof. The initial distribution is concentrated at x. Since the actor terminates with zero reward at both states,

$$V^{\pi_A}(x) = V^{\pi_A}(y) = 0.$$

Consequently,

$$Q^{\pi_A}(x, a_F) - Q^{\pi_A}(x, a_A) - c_x = -\kappa - c_x < 0.$$

A cost-adjusted selector using exact Q^{π_A} rejects the flow action at x, independently of its opportunity variable. It obtains zero return.

For the full-opportunity recursive optimum, $R > c_y$ implies

$$V_1^*(y) = R - c_y.$$

Its cost-adjusted difference at x is

$$\begin{aligned}\Delta^*(x) &= -\kappa - c_x + \gamma(R - c_y) \\ &\geq -\kappa - c_x + \gamma p(R - c_y) > 0.\end{aligned}$$

The recursive selector therefore accepts the flow action at both states.

Under independent deployment opportunities with probability p,

$$V_c^{\widehat{\pi}_p}(y) = p(R - c_y),$$

and

$$\begin{aligned}J_c(\widehat{\pi}_p) &= p[-\kappa - c_x + \gamma V_c^{\widehat{\pi}_p}(y)] \\ &= p[-\kappa - c_x + \gamma p(R - c_y)] > 0.\end{aligned}$$

This proves the strict comparison.

A compatible numerical instance uses $c_x = c_y = 0.125$, candidate distance $d = 0.1$, $\gamma = 0.9$, $p = 0.5$, $\kappa = 0.2$, and $R = 1$. The net return is 0.034375, and the environment return is 0.125. These values describe the constructed MDP, not an additional benchmark. □

Exact FQE remains a valid improvement baseline. At $t = 0$, Equation (4) is the evaluation operator of π_A, so $q_0^* = Q^{\pi_A}$. Let

$$h_0(s, K) = e(s, K)\mathbf{1}\{Q^{\pi_A}(s, a_F) - Q^{\pi_A}(s, a_A) > c(s, K)\}.$$

Its net local advantage relative to the actor is nonnegative:

$$p\mathbb{E}_K h_0 \left[Q^{\pi_A}(s, a_F) - Q^{\pi_A}(s, a_A) - c\right] \geq 0.$$

Applying Equation (11), proved below, gives

$$J_c(\pi_{p,h_0}) \geq J(\pi_A).$$

The distinction in Proposition 2 is therefore not that exact FQE is unsafe, but that one-step improvement need not attain the best value available through repeated selection.

B.3 Proof of Theorem 1

Proof. Let V_1^* be the exact full-opportunity cost-adjusted state value. The relation $q_1^*(s, u) = r(s, u) + \gamma \mathbb{E}_{s'|s,u} V_1^*(s')$ implies

$$(\mathcal{B}_c^{\widehat{\pi}_p} V_1^*)(s) = \mathbb{E}_K \left[q_1^*(s, a_A) + pe\widehat{I}\Delta^*\right].$$

The optimal state value is

$$V_1^*(s) = \mathbb{E}_K \left[q_1^*(s, a_A) + e[\Delta^*]_+\right].$$

Define the one-step loss

$$\ell_p(s) = V_1^*(s) - (\mathcal{B}_c^{\widehat{\pi}_p} V_1^*)(s) = \mathbb{E}_K e \left([\Delta^*]_+ - p\widehat{I}\Delta^*\right).$$

For any real x, $I^* = \mathbf{1}\{x > 0\}$ and $\widehat{I} \in \{0, 1\}$ satisfy

$$[x]_+ - \widehat{I}x = |x|\mathbf{1}\{\widehat{I} \neq I^*\}.$$

This follows by considering $x > 0$, $x < 0$, and $x = 0$ separately. Hence

$$[\Delta^*]_+ - p\widehat{I}\Delta^* = (1-p)[\Delta^*]_+ + p|\Delta^*|\mathbf{1}\{\widehat{I} \neq I^*\}.$$

Let P_p be the transition kernel induced by $\widehat{\pi}_p$. Subtracting its evaluation equation yields

$$V_1^* - V_c^{\widehat{\pi}_p} = \ell_p + \gamma P_p \left(V_1^* - V_c^{\widehat{\pi}_p}\right).$$

Iteration gives

$$V_1^* - V_c^{\widehat{\pi}_p} = \sum_{k=0}^{\infty} \gamma^k P_p^k \ell_p.$$

The series converges uniformly because ℓ_p is bounded and $\gamma < 1$. Averaging under μ_0 and using the definition of $d^{\widehat{\pi}_p}$ gives

$$J_c(\pi_1^*) - J_c(\widehat{\pi}_p) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\widehat{\pi}_p}} \ell_p(s),$$

which is Equation (8).

Moreover,

$$\Delta_{\widehat{q}} - \Delta^* = E_{\text{pair}}.$$

If $\Delta^* > 0$ and $\widehat{I} = 0$, then $\Delta_{\widehat{q}} \leq 0$, so $|E_{\text{pair}}| \geq \Delta^*$. If $\Delta^* \leq 0$ and $\widehat{I} = 1$, then $\Delta_{\widehat{q}} > 0$, so $|E_{\text{pair}}| \geq |\Delta^*|$. This includes the zero-gap case and proves the error bound. □

The decomposition also gives a direct environment-return comparison. Let $B^* = J_c(\pi_1^*) - J(\pi_A) \geq 0$. Then

$$\begin{aligned} J(\widehat{\pi}_p) - J(\pi_A) = B^* + C(\widehat{\pi}_p) \\ - \frac{1}{1-\gamma}\mathbb{E}_{d^{\widehat{\pi}_p},K}\left[e\left((1-p)[\Delta^*]_+ + p|\Delta^*|\mathbf{1}\{\widehat{I} \neq I^*\}\right)\right]. \end{aligned} \tag{10}$$

Thus the exact recursive surplus and the cost added back to environment return are offset by skipped selections and ranking errors. Replacing $|\Delta^*|\mathbf{1}\{\widehat{I} \neq I^*\}$ by $|E_{\text{pair}}|\mathbf{1}\{\widehat{I} \neq I^*\}$ provides a sufficient lower bound.

B.4 Exact Comparison with the Actor

Fix the fitted score and write $h = e\widehat{I}$ and

$$G(s, K) = Q^{\pi_A}(s, a_F) - Q^{\pi_A}(s, a_A).$$

The unconditional actor-candidate marginal is π_A, so

$$\mathbb{E}_K Q^{\pi_A}(s, a_A(s, K)) = V^{\pi_A}(s).$$

It follows that

$$\begin{aligned} &\mathbb{E}_{K,Z}\left[Q^{\pi_A}(s, a_{\text{out}}(s, K, Z)) - V^{\pi_A}(s)\right] \\ &\quad = p\mathbb{E}_K[hG]. \end{aligned}$$

This cancellation uses the full actor-candidate distribution, not the actor distribution conditioned on rejection.

Telescoping along a trajectory of $\widehat{\pi}_p$ gives

$$\begin{aligned} J(\widehat{\pi}_p) - J(\pi_A) &= \mathbb{E}_{\widehat{\pi}_p} \sum_{k\geq 0} \gamma^k \left[r_k + \gamma V^{\pi_A}(s_{k+1}) - V^{\pi_A}(s_k)\right] \\ &= \frac{p}{1-\gamma}\mathbb{E}_{d^{\widehat{\pi}_p},K}[hG]. \end{aligned}$$

The terminal term vanishes because V^{π_A} is bounded. The expected replacement cost is

$$C(\widehat{\pi}_p) = \frac{p}{1-\gamma}\mathbb{E}_{d^{\widehat{\pi}_p},K}[hc].$$

Subtracting it yields

$$J_c(\widehat{\pi}_p) - J(\pi_A) = \frac{p}{1-\gamma}\mathbb{E}_{s\sim d^{\widehat{\pi}_p}} g(s), \qquad g(s) = \mathbb{E}_K[h(G-c)]. \tag{11}$$

For an actor-relative action-gap condition, define

$$E_A(s,K) = [\widehat{q}(s,a_F) - \widehat{q}(s,a_A)] - G(s,K).$$

Whenever $h = 1$, the fitted difference exceeds c, giving

$$hG = h[\widehat{q}(s,a_F) - \widehat{q}(s,a_A) - E_A] \geq h(c - |E_A|).$$

Therefore

$$J(\widehat{\pi}_p) - J(\pi_A) \geq \frac{p}{1-\gamma}\mathbb{E}_{d^{\widehat{\pi}_p},K}[h(c-|E_A|)].$$

For $p > 0$, the condition

$$\mathbb{E}_{d^{\widehat{\pi}_p},K}[hc] > \mathbb{E}_{d^{\widehat{\pi}_p},K}[h|E_A|]$$

is sufficient for strict improvement in environment return. This condition concerns true actor-relative differences at states visited by the deployed policy. Empirical TD loss and twin-critic disagreement do not directly measure E_A.

B.5 Proof of Theorem 2

Proof. Let π_{sel} execute every accepted replacement. With the fitted score fixed,

$$\widehat{\pi}_p = (1-p)\pi_A + p\pi_{\mathrm{sel}}$$

holds at every state. Let $P_A, P_{\mathrm{sel}}, P_p$ be the corresponding transition kernels, and let $d_A = d^{\pi_A}$, $d_p = d^{\widehat{\pi}_p}$. Using row-measure notation,

$$P_p = (1-p)P_A + pP_{\mathrm{sel}},$$

$$d_p = (1-\gamma)\mu_0 + \gamma d_p P_p, \qquad d_A = (1-\gamma)\mu_0 + \gamma d_A P_A.$$

Subtracting these equations gives

$$(d_p - d_A)(I - \gamma P_p) = \gamma d_A(P_p - P_A),$$

and hence

$$d_p - d_A = \gamma d_A(P_p - P_A)(I-\gamma P_p)^{-1}.$$

For a fixed state,

$$P_{\mathrm{sel}}(s,\cdot) - P_A(s,\cdot) = \mathbb{E}_K h(s,K)\left[P(\cdot \mid s,a_F) - P(\cdot \mid s,a_A)\right].$$

Use the total-variation norm of a signed measure, $\|\nu\|_{\mathrm{TV}} = |\nu|(\mathcal{S})$. In this convention, the norm of the difference of two probability measures is at most two. Thus

$$\|P_{\mathrm{sel}}(s,\cdot) - P_A(s,\cdot)\|_{\mathrm{TV}} \leq 2\mathbb{E}_K h(s,K).$$

Markov kernels are nonexpansive in this norm, and

$$(I-\gamma P_p)^{-1} = \sum_{k\geq 0}\gamma^k P_p^k$$

has operator norm at most $(1-\gamma)^{-1}$. Consequently,

$$\|d_p - d_A\|_{\mathrm{TV}} \leq \frac{2\gamma p}{1-\gamma}\mathbb{E}_{d_A,K}h = \frac{2\gamma p\bar{h}}{1-\gamma}.$$

Since $d_p - d_A$ has total mass zero, centering g at $(\sup_s g(s) + \inf_s g(s))/2$ yields

$$\begin{aligned}\left|\mathbb{E}_{d_p} g - \mathbb{E}_{d_A} g\right| &\leq \frac{S_g}{2} \|d_p - d_A\|_{\mathrm{TV}} \\ &\leq \frac{\gamma p \bar{h} S_g}{1-\gamma}.\end{aligned}$$

Using Equation (11),

$$\begin{aligned}J_c(\widehat{\pi}_p) - J(\pi_A) &= \frac{p}{1-\gamma} \mathbb{E}_{d_p} g \\ &\geq \frac{p}{1-\gamma} \left[A - \frac{\gamma p \bar{h} S_g}{1-\gamma} \right].\end{aligned}$$

This proves Equation (9). When $A > 0$, the stated conditions make the bracket strictly positive. The case $\bar{h} S_g = 0$ follows directly. No independence between visited states is assumed. □

B.6 Why Exact Recursive Values Do Not Guarantee Improvement After Probabilistic Rejection

Consider the deterministic two-state construction from Appendix B.2, with $\gamma = 0.9$, $R = 1$, and first flow reward -0.6. Use $c_x = c_y = 0.125$ and $d = 0.1$, which satisfy the experimental cost and eligibility rules. The exact full-opportunity values satisfy

$$V_1^*(y) = 0.875, \qquad q_1^*(x, a_F) = -0.6 + 0.9(0.875) = 0.1875,$$

$$\Delta^*(x) = 0.1875 - 0.125 = 0.0625 > 0.$$

The exact recursive selector accepts both flow actions. With full execution, its environment return is

$$-0.6 + 0.9 = 0.3 > 0.$$

With $p = 0.5$, the same exact selector instead obtains

$$J(\widehat{\pi}_{0.5}) = 0.5[-0.6 + 0.9(0.5)] = -0.075 < 0 = J(\pi_A).$$

Thus exact q_1^* does not imply actor-relative improvement after changing the execution probability. The positive condition in Proposition 2 and the conditions in Theorem 2 are not satisfied automatically.

B.7 Learning Error and the Scope of the Guarantees

If $\|\widehat{q} - q_1^*\|_\infty \leq \epsilon$, then

$$|E_{\mathrm{pair}}| \leq 2\epsilon.$$

If the true uniform Bellman residual

$$\beta = \|\widehat{q} - \mathcal{T}_1 \widehat{q}\|_\infty$$

is available, contraction implies

$$\begin{aligned}\|\widehat{q} - q_1^*\|_\infty &\leq \|\widehat{q} - \mathcal{T}_1 \widehat{q}\|_\infty + \|\mathcal{T}_1 \widehat{q} - \mathcal{T}_1 q_1^*\|_\infty \\ &\leq \beta + \gamma \|\widehat{q} - q_1^*\|_\infty.\end{aligned}$$

Rearranging gives

$$\|\widehat{q} - q_1^*\|_\infty \leq \frac{\beta}{1-\gamma}, \qquad |E_{\mathrm{pair}}| \leq \frac{2\beta}{1-\gamma}.$$

These bounds control the ranking-error term in Equation (8). They do not remove its skipped-selection term when $p < 1$.

The empirical squared TD loss is evaluated on logged samples against a moving target network. It is not β. Relating finite-data fitting error to the population operator requires additional coverage and function-approximation conditions (Jin et al., 2021; Xie et al., 2021). The neural-network experiments do not establish such a uniform error certificate. Likewise, the $50{,}000$ Adam updates are not $50{,}000$ exact Bellman iterations.

B.8 Finite Evaluation Prefixes and the Theoretical Objective

Let T be a fixed evaluation horizon. Pad a true absorbing termination with zero reward and zero replacement cost, and define

$$J_{c,T}(\pi) = \mathbb{E}_\pi \sum_{k=0}^{T-1} \gamma^k (r_k - c_k H_k), \qquad J_{A,T} = \mathbb{E}_{\pi_A} \sum_{k=0}^{T-1} \gamma^k r_k.$$

Splitting each infinite discounted sum at T and conditioning on s_T gives

$$J_c(\pi) = J_{c,T}(\pi) + \gamma^T \mathbb{E}_\pi V_c^\pi(s_T),$$

$$J(\pi_A) = J_{A,T} + \gamma^T \mathbb{E}_{\pi_A} V^{\pi_A}(s_T).$$

Subtracting yields

$$\begin{aligned} J_c(\pi) - J(\pi_A) = J_{c,T}(\pi) - J_{A,T} \\ + \gamma^T \left[\mathbb{E}_\pi V_c^\pi(s_T) - \mathbb{E}_{\pi_A} V^{\pi_A}(s_T) \right]. \end{aligned} \tag{12}$$

Under Assumption 1, the absolute boundary term is at most

$$\frac{\gamma^T (2R_{\max} + c_{\max})}{1-\gamma}.$$

Indeed,

$$\|V_c^\pi\|_\infty \le \frac{R_{\max} + c_{\max}}{1-\gamma}, \qquad \|V^{\pi_A}\|_\infty \le \frac{R_{\max}}{1-\gamma}.$$

A dataset boundary with $m_i = 1$ is not Bellman termination. If a time limit is instead part of the task's terminal condition, time must be included in the state and the value after that limit is zero. These two conventions cannot be mixed. Finite-horizon success rates and recorded reward prefixes are empirical outcomes, not direct measurements of the infinite-horizon bounds or their unknown error terms.

C Implementation and Evaluation Details

C.1 Pretrained Actor–Flow Inputs

BACI is applied to pretrained actor–flow pairs produced by GFP or FAC. For both input methods, the actor, flow policy, and original critic are trained before BACI refinement and remain fixed while the additional decision critic is fitted. BACI therefore changes neither action generator during its 50K decision-critic updates.

For reference, standard conditional flow matching (Lipman et al., 2023) fits a velocity field using

$$\mathcal{L}_{\mathrm{FM}}(\omega) = \mathbb{E}_{(s,a)\sim\mathcal{D},\, z\sim\mathcal{N}(0,I),\, \xi\sim U[0,1]} \left\| v_\omega(s, (1-\xi)z + \xi a, \xi) - (a - z) \right\|_2^2. \tag{13}$$

Here z is Gaussian noise and ξ denotes flow interpolation time. A common actor objective combines value maximization with alignment to the flow endpoint,

$$\mathcal{L}_A(\theta) = \mathbb{E}_{s,z} \left[-Q_\phi(s, A_\theta(s,z)) + \beta \left\| A_\theta(s,z) - \mathrm{sg}(F_\omega(s,z)) \right\|_2^2 \right], \tag{14}$$

where F_ω denotes the numerically integrated flow endpoint and $\beta \ge 0$ controls the alignment term. These equations summarize the inherited actor–flow structure and are not additional terms in the BACI decision-critic objective.

GFP couples a value-aware flow policy with its one-step actor and critic (Tiofack et al., 2026). FAC trains a flow-based behavior proxy and uses it to regularize critic learning (Chae et al., 2026). Their original training procedures are retained when constructing the pretrained actor–flow pairs. BACI subsequently applies the same decision-critic training and action-selection equations to either pair.

C.2 Implementation Details

For every decision, the actor and flow policy use independent random inputs. The actor produces one action with a single network evaluation, while the flow proposal is generated with the integration procedure of the corresponding pretrained model. Both actions are clipped to $[-1, 1]^{d_a}$. Fresh random inputs are drawn at every state.

The BACI decision critics copy the parameters of the pretrained target critics and initialize new Adam optimizer states. The pretrained actor, flow policy, and original critic are not updated during BACI refinement. For each generated candidate action, the two target decision-critic heads are evaluated separately and their minimum is used in the Bellman target. The regression loss is evaluated at the logged dataset action, whereas generated actions appear only in the next-state backup, as specified in Equation (6).

After every decision-critic update, the target parameters are moved toward the updated online parameters using Polyak averaging. The target critics are used to construct Bellman targets, while the online decision critics are used for deployment. A flow proposal is eligible only when both candidate actions are finite, distinct, and satisfy the distance constraint in Equation (3). Rejected proposals return the actor action already generated at that state.

Table 3: Shared BACI decision-critic and action-selection settings. Actor and flow architectures, discount factors, and flow integration steps are inherited from the corresponding pretrained GFP or FAC model.

Quantity	Setting
Decision-critic updates / batch size	50,000 / 256
Adam learning rate / moments / ϵ	3×10^{-4} / $(0.9, 0.999)$ / 10^{-8}
Target update	$\bar{\psi} \leftarrow 0.995\bar{\psi} + 0.005\psi$
Decision-critic initialization	Pretrained target-critic parameters; new Adam state
Main training / deployment probability	$t = 1, p = 0.5$
Actor–flow distance	$\|a_F - a_A\|_2^2/(4d_a)$
Replacement cost	$0.10 + 0.25d$
Maximum eligible distance	0.15
Flow proposals per state	One
Fallback after rejection	Previously generated actor action
Pretrained-network updates during BACI training	None

C.3 Matched Control Variants

The supplementary controls modify only the component required for the corresponding comparison.

Actor and **Flow** execute the pretrained actor or flow policy alone for the full episode. **Q-select** applies the BACI candidate constraints and deployment rule using the original target critic without fitting an additional decision critic. **FQE** uses the same decision-critic initialization, architecture, optimizer, minibatches, and 50K update budget as BACI, but sets $t = 0$ so that the Bellman target evaluates future actor actions. The main BACI configuration uses $t = 1$. Deployment with $p = 1$ reuses the same fitted critic as its corresponding $p = 0.5$ variant and changes only the probability of executing an accepted flow proposal.

The cost–distance ablation keeps the pretrained actor and flow policy fixed. Removing the cost sets $c_0 = \lambda_d = 0$ during both critic training and deployment. Removing the distance restriction eliminates the hard radius test while retaining action clipping, finite-candidate checks, distinct actions, and the replacement cost. Removing both applies both modifications.

C.4 Task Scope and Evaluation Protocol

The main benchmark results and the additional diagnostic controls use different statistical protocols and are reported separately.

Main OGBench evaluation. The OGBench task settings follow the rows reported in Table 1. Each reported BACI category contains five goal-specific policies. For every training seed, each goal is evaluated over 100 episodes. The five goal scores are averaged to obtain a category score, and the reported BACI value is the mean $\pm$ sample standard deviation of that category score over 10 independent training seeds.

Main D4RL evaluation. The D4RL task settings follow Table 2. BACI is evaluated over 10 independent training seeds. For every seed, AntMaze tasks use 50 evaluation episodes, while MuJoCo and Adroit tasks use 10 episodes. GFP-trained and FAC-trained actor–flow pairs cover the task subsets shown in the table; results from one input pair are not substituted for the other.

Additional matched controls. The detailed control tables later in the appendix use their explicitly stated protocols. Unless a table states otherwise, the per-goal diagnostic controls use the fixed seed-zero pretrained endpoints with 25 paired evaluation episodes. These experiments analyze action selection for fixed trained models.

All evaluations use fixed checkpoints selected by training update count rather than evaluation performance. Changing the physical machine used to execute an experiment does not create an additional training replicate.

C.5 Published Baselines and Missing Entries

Published IQL, ReBRAC, FQL, GFP, and FAC values in the main benchmark tables are taken from the cited source papers and retain the statistics and evaluation protocols reported there (Tiofack et al., 2026; Chae et al., 2026). They are not reinterpreted as matched reruns of our protocol.

In Tables 1 and 2, "-" indicates that the corresponding task setting is not reported for that input method in its source paper; the corresponding BACI experiment is therefore not included. Cross-paper results are used for benchmark comparison, whereas method-level attribution relies on matched controls using the same pretrained actor–flow pair.

C.6 Fixed Design Choices and Computation

The main selection parameters $(r_{\max}, c_0, \lambda_d) = (0.15, 0.10, 0.25)$ and $(t, p) = (1, 0.5)$ are held fixed across the reported main evaluations. The supplementary controls vary p or remove individual selection constraints without changing the pretrained actor–flow pair.

The replacement cost is measured in reward units, whereas the normalized actor–flow distance is dimensionless. Under a positive reward scaling $r' = \alpha r$, the corresponding value and replacement-cost quantities must be scaled by the same factor to preserve the exact selection comparison. The distance constraint itself is not interpreted as an estimate of behavior support.

The 50K decision-critic updates are additional to the pretrained model's training. Deployment generates one actor action and one flow proposal and evaluates both using the decision critic. Because candidates are generated before the execution-probability draw, reducing p does not proportionally reduce action-generation cost.

C.7 Statistical Aggregation

For an OGBench category j, goal g, and training seed s, let $X_{j,g,s}$ denote the success rate obtained from the 100 evaluation episodes. The category score for seed s is

$$Z_{j,s} = \frac{1}{5} \sum_{g=1}^{5} X_{j,g,s}.$$

With $S = 10$ independent training seeds, the reported category mean and sample standard deviation are

$$\overline{Z}_j = \frac{1}{S} \sum_{s=1}^{S} Z_{j,s}, \qquad \mathrm{SD}(Z_j) = \sqrt{\frac{1}{S-1} \sum_{s=1}^{S} (Z_{j,s} - \overline{Z}_j)^2}.$$

For D4RL, the per-seed task score is first averaged over that seed's evaluation episodes: 50 episodes for AntMaze and 10 for MuJoCo or Adroit. The table then reports the mean and sample standard deviation of these per-seed task scores over the same 10 independent training seeds.

The detailed seed-zero control tables and their paired episode analyses later in the appendix address variability for fixed trained models and should not be interpreted as estimates of training-seed variability. Missing task settings are not dropped to improve an aggregate, and OGBench success rates are not combined with D4RL normalized returns.

D Complete Measured Actor–Flow Results

The current empirical scope uses native one-step actors as the reference in every reported run. GFP and FAC provide independently trained policy pairs. The following tables retain all 50 OGBench goals per pair and all 44 measured D4RL input–task combinations, including zero-success goals. All measured values use training seed zero and 25 paired episodes.

Table 4: All available raw-actor controls over complete five-goal categories. Values are success percentages, training seed zero, 25 paired episodes per goal. Actor/Flow run the corresponding native policy alone; Q selects with the native critic without extra fitting; $\text{FQE}_{.5}$ and FQE_1 use the same fitted reference-evaluation critic; $\text{BACI}_{.5}$ and BACI_1 use the same recursive critic. Subscripts are deployment probabilities.

Category	Actor	Flow	Q	$\text{FQE}_{.5}$	$\text{BACI}_{.5}$	BACI_1	FQE_1
GFP reference							
antmaze-large-navigate	96.0	90.4	95.2	96.0	97.6	95.2	94.0
antmaze-giant-navigate	8.0	0.0	9.6	7.2	28.8	40.8	8.8
humanoid-medium-navigate	72.8	40.8	77.6	75.2	78.4	72.0	76.4
humanoid-large-navigate	30.4	4.8	31.2	33.6	34.4	48.0	32.4
antsoccer-navigate	73.6	12.8	63.2	67.2	70.4	63.2	69.6
cube-single-play	100.0	32.8	100.0	100.0	100.0	97.6	100.0
cube-double-play	40.8	5.6	48.0	48.8	48.0	46.4	47.6
puzzle-3×3-play	20.0	20.0	19.2	20.0	20.0	19.2	20.0
puzzle-4×4-play	22.4	10.4	24.0	24.0	25.6	29.6	26.4
scene-play	52.8	59.2	52.0	55.2	54.4	56.0	54.0
FAC reference							
antmaze-large-navigate	90.4	2.4	80.8	85.6	77.6	81.6	84.4
antmaze-giant-navigate	23.2	0.0	28.0	34.4	35.2	38.4	34.8
humanoid-medium-navigate	62.4	3.2	68.0	60.8	67.2	69.6	66.4
humanoid-large-navigate	5.6	0.0	2.4	8.8	5.6	7.2	7.2
antsoccer-navigate	68.0	4.0	62.4	52.0	64.0	63.2	57.6
cube-single-play	96.0	12.8	96.0	97.6	96.0	92.0	95.2
cube-double-play	20.0	4.0	21.6	23.2	27.2	27.2	26.8
puzzle-3×3-play	100.0	3.2	100.0	100.0	99.2	100.0	100.0
puzzle-4×4-play	26.4	0.0	30.4	35.2	32.8	36.0	36.0
scene-play	63.2	8.0	62.4	64.0	64.0	68.0	64.0

Table 5: All extraction controls on GFP references, part 1/2. Each goal uses its own fixed seed-zero native endpoint and 25 common evaluation initializations. Labels follow Table 4.

Goal	Actor	Flow	Q	$FQE_{.5}$	$BACI_{.5}$	$BACI_1$	FQE_1
antmaze-large-navigate							
task1	88.0	88.0	96.0	92.0	96.0	96.0	86.0
task2	96.0	88.0	92.0	96.0	96.0	92.0	90.0
task3	100.0	96.0	100.0	100.0	100.0	100.0	100.0
task4	96.0	88.0	92.0	92.0	96.0	88.0	94.0
task5	100.0	92.0	96.0	100.0	100.0	100.0	100.0
antmaze-giant-navigate							
task1	0.0	0.0	0.0	0.0	0.0	16.0	0.0
task2	0.0	0.0	4.0	0.0	60.0	92.0	0.0
task3	28.0	0.0	44.0	24.0	56.0	64.0	30.0
task4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
task5	12.0	0.0	0.0	12.0	28.0	32.0	14.0
humanoid-medium-navigate							
task1	84.0	20.0	100.0	88.0	96.0	84.0	94.0
task2	100.0	76.0	96.0	96.0	100.0	88.0	94.0
task3	84.0	40.0	92.0	96.0	96.0	88.0	96.0
task4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
task5	96.0	68.0	100.0	96.0	100.0	100.0	98.0
humanoid-large-navigate							
task1	92.0	12.0	80.0	84.0	84.0	88.0	86.0
task2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
task3	0.0	4.0	0.0	0.0	0.0	48.0	0.0
task4	0.0	0.0	16.0	28.0	8.0	28.0	26.0
task5	60.0	8.0	60.0	56.0	80.0	76.0	50.0
antsoccer-navigate							
task1	96.0	16.0	88.0	80.0	84.0	92.0	86.0
task2	100.0	32.0	100.0	100.0	96.0	96.0	100.0
task3	76.0	8.0	60.0	60.0	64.0	60.0	62.0
task4	52.0	4.0	48.0	60.0	72.0	24.0	64.0
task5	44.0	4.0	20.0	36.0	36.0	44.0	36.0

Table 6: All extraction controls on GFP references, part 2/2. Each goal uses its own fixed seed-zero native endpoint and 25 common evaluation initializations. Labels follow Table 4.

Goal	Actor	Flow	Q	FQE$_{.5}$	BACI$_{.5}$	BACI$_1$	FQE$_1$
cube-single-play							
task1	100.0	32.0	100.0	100.0	100.0	100.0	100.0
task2	100.0	20.0	100.0	100.0	100.0	100.0	100.0
task3	100.0	32.0	100.0	100.0	100.0	100.0	100.0
task4	100.0	44.0	100.0	100.0	100.0	96.0	100.0
task5	100.0	36.0	100.0	100.0	100.0	92.0	100.0
cube-double-play							
task1	60.0	20.0	76.0	88.0	84.0	76.0	86.0
task2	44.0	4.0	56.0	28.0	52.0	52.0	32.0
task3	32.0	0.0	36.0	52.0	40.0	32.0	48.0
task4	4.0	0.0	12.0	4.0	8.0	8.0	6.0
task5	64.0	4.0	60.0	72.0	56.0	64.0	66.0
puzzle-3×3-play							
task1	96.0	56.0	96.0	92.0	92.0	96.0	100.0
task2	0.0	8.0	0.0	0.0	0.0	0.0	0.0
task3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
task4	4.0	20.0	0.0	4.0	4.0	0.0	0.0
task5	0.0	16.0	0.0	4.0	4.0	0.0	0.0
puzzle-4×4-play							
task1	48.0	16.0	44.0	44.0	48.0	56.0	32.0
task2	8.0	8.0	8.0	12.0	8.0	8.0	8.0
task3	28.0	24.0	52.0	48.0	44.0	68.0	80.0
task4	12.0	0.0	4.0	12.0	20.0	8.0	8.0
task5	16.0	4.0	12.0	4.0	8.0	8.0	4.0
scene-play							
task1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
task2	92.0	100.0	100.0	100.0	96.0	100.0	96.0
task3	68.0	96.0	60.0	76.0	76.0	80.0	74.0
task4	4.0	0.0	0.0	0.0	0.0	0.0	0.0
task5	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 7: All extraction controls on FAC references, part 1/2. Each goal uses its own fixed seed-zero native endpoint and 25 common evaluation initializations. Labels follow Table 4.

Goal	Actor	Flow	Q	$FQE_{.5}$	$BACI_{.5}$	$BACI_1$	FQE_1
antmaze-large-navigate							
task1	84.0	0.0	88.0	96.0	88.0	84.0	94.0
task2	88.0	0.0	76.0	92.0	72.0	80.0	94.0
task3	96.0	8.0	96.0	88.0	84.0	92.0	84.0
task4	88.0	0.0	52.0	56.0	56.0	64.0	58.0
task5	96.0	4.0	92.0	96.0	88.0	88.0	92.0
antmaze-giant-navigate							
task1	0.0	0.0	0.0	0.0	4.0	0.0	0.0
task2	24.0	0.0	48.0	52.0	72.0	56.0	54.0
task3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
task4	36.0	0.0	32.0	40.0	24.0	56.0	46.0
task5	56.0	0.0	60.0	80.0	76.0	80.0	74.0
humanoid-medium-navigate							
task1	68.0	4.0	92.0	36.0	68.0	80.0	60.0
task2	100.0	0.0	96.0	84.0	92.0	96.0	92.0
task3	52.0	12.0	52.0	80.0	76.0	76.0	78.0
task4	0.0	0.0	0.0	4.0	0.0	0.0	2.0
task5	92.0	0.0	100.0	100.0	100.0	96.0	100.0
humanoid-large-navigate							
task1	4.0	0.0	0.0	8.0	12.0	20.0	12.0
task2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
task3	24.0	0.0	12.0	36.0	16.0	16.0	24.0
task4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
task5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
antsoccer-navigate							
task1	80.0	8.0	64.0	64.0	80.0	88.0	68.0
task2	100.0	12.0	96.0	88.0	92.0	100.0	90.0
task3	60.0	0.0	60.0	44.0	60.0	48.0	60.0
task4	52.0	0.0	48.0	40.0	44.0	32.0	48.0
task5	48.0	0.0	44.0	24.0	44.0	48.0	22.0

Table 8: All extraction controls on FAC references, part 2/2. Each goal uses its own fixed seed-zero native endpoint and 25 common evaluation initializations. Labels follow Table 4.

Goal	Actor	Flow	Q	$FQE_{.5}$	$BACI_{.5}$	$BACI_1$	FQE_1
cube-single-play							
task1	96.0	12.0	96.0	96.0	96.0	96.0	94.0
task2	96.0	4.0	96.0	100.0	100.0	96.0	100.0
task3	100.0	20.0	100.0	100.0	100.0	96.0	100.0
task4	100.0	12.0	100.0	100.0	96.0	92.0	100.0
task5	88.0	16.0	88.0	92.0	88.0	80.0	82.0
cube-double-play							
task1	44.0	20.0	36.0	56.0	56.0	44.0	56.0
task2	4.0	0.0	32.0	20.0	32.0	40.0	24.0
task3	16.0	0.0	16.0	20.0	20.0	24.0	20.0
task4	4.0	0.0	0.0	4.0	0.0	0.0	6.0
task5	32.0	0.0	24.0	16.0	28.0	28.0	32.0
puzzle-3×3-play							
task1	100.0	12.0	100.0	100.0	100.0	100.0	100.0
task2	100.0	4.0	100.0	100.0	100.0	100.0	100.0
task3	100.0	0.0	100.0	100.0	100.0	100.0	100.0
task4	100.0	0.0	100.0	100.0	100.0	100.0	100.0
task5	100.0	0.0	100.0	100.0	96.0	100.0	100.0
puzzle-4×4-play							
task1	44.0	0.0	48.0	48.0	48.0	60.0	36.0
task2	0.0	0.0	4.0	4.0	0.0	0.0	0.0
task3	68.0	0.0	68.0	88.0	80.0	88.0	80.0
task4	20.0	0.0	32.0	36.0	36.0	24.0	64.0
task5	0.0	0.0	0.0	0.0	0.0	8.0	0.0
scene-play							
task1	100.0	28.0	100.0	100.0	100.0	100.0	100.0
task2	100.0	0.0	92.0	96.0	96.0	96.0	96.0
task3	100.0	0.0	100.0	100.0	96.0	100.0	100.0
task4	16.0	12.0	20.0	24.0	28.0	44.0	24.0
task5	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 9: All D4RL extraction modes with GFP-trained inputs. The complete 27-task union is retained: 21 task rows are measured. Values are normalized returns for training seed zero, 25 paired episodes, final native 1M weights, and 50K updates for fitted critics. Actor/Flow use the native policies, Q-select uses the native target critic, $\text{FQE}_{.5}$ uses reference-only evaluation, and $\text{BACI}_{.5}$/BACI_1 share the recursive critic. Subscripts are deployment probabilities.

Task	Actor	Flow	Q-select	$\text{FQE}_{.5}$	$\text{BACI}_{.5}$	BACI_1
halfcheetah-mr	45.02	42.44	45.43	45.87	45.83	46.59
hopper-mr	29.30	26.37	24.96	26.30	25.88	25.47
walker2d-mr	25.32	18.13	27.46	33.58	24.97	26.09
antmaze-umaze	100.00	88.00	96.00	100.00	100.00	96.00
antmaze-umaze-diverse	88.00	84.00	88.00	92.00	92.00	84.00
antmaze-m-play	40.00	52.00	40.00	56.00	48.00	68.00
antmaze-m-diverse	12.00	28.00	20.00	8.00	12.00	24.00
antmaze-large-play	84.00	72.00	88.00	84.00	84.00	84.00
antmaze-large-diverse	80.00	80.00	80.00	88.00	76.00	80.00
pen-human	73.24	82.37	76.47	76.93	71.88	69.30
pen-cloned	81.03	82.69	82.88	74.11	87.85	91.33
pen-expert	139.91	125.79	140.18	148.60	148.41	152.30
door-human	0.03	2.25	0.04	0.02	0.06	2.63
door-cloned	−0.01	−0.11	0.03	−0.00	0.09	0.36
door-expert	104.71	104.53	105.09	104.61	105.07	105.11
hammer-human	4.49	1.26	9.45	5.57	5.22	5.30
hammer-cloned	12.51	4.31	9.78	14.30	8.59	6.57
hammer-expert	121.32	114.08	121.82	128.39	128.47	129.46
relocate-human	0.28	0.01	0.06	1.13	0.93	0.05
relocate-cloned	0.82	0.09	0.81	0.81	0.92	0.48
relocate-expert	109.14	106.16	108.39	108.21	108.21	108.24

Table 10: All D4RL extraction modes with FAC-trained inputs. The complete 27-task union is retained: 23 task rows are measured. Values are normalized returns for training seed zero, 25 paired episodes, final native 1M weights, and 50K updates for fitted critics. Actor/Flow use the native policies, Q-select uses the native target critic, $\text{FQE}_{.5}$ uses reference-only evaluation, and $\text{BACI}_{.5}$/BACI_1 share the recursive critic. Subscripts are deployment probabilities.

Task	Actor	Flow	Q-select	$\text{FQE}_{.5}$	$\text{BACI}_{.5}$	BACI_1
halfcheetah-m	64.88	41.82	64.41	64.56	64.13	63.28
hopper-m	77.19	42.84	76.96	81.41	78.86	78.80
walker2d-m	85.42	54.57	85.14	85.52	85.50	86.20
halfcheetah-mr	53.50	34.78	55.35	53.09	55.38	55.48
hopper-mr	95.30	21.82	95.93	97.18	97.33	98.85
walker2d-mr	56.26	18.51	71.78	73.79	80.67	73.16
halfcheetah-me	102.15	54.27	100.13	94.25	102.13	99.28
hopper-me	104.33	51.92	108.37	106.48	102.69	111.95
walker2d-me	108.38	82.23	108.22	108.33	108.20	108.44
antmaze-umaze	96.00	52.00	100.00	96.00	100.00	100.00
antmaze-umaze-diverse	92.00	68.00	100.00	96.00	96.00	100.00
antmaze-m-play	76.00	0.00	80.00	88.00	80.00	88.00
antmaze-m-diverse	72.00	0.00	60.00	88.00	72.00	88.00
antmaze-large-play	84.00	0.00	80.00	80.00	84.00	92.00
antmaze-large-diverse	88.00	0.00	92.00	92.00	88.00	80.00
pen-human	70.65	51.87	72.29	68.04	68.69	73.38
pen-cloned	101.47	86.37	104.48	105.14	108.52	111.52
door-human	8.64	0.01	3.68	6.89	6.16	2.62
door-cloned	8.08	−0.08	9.44	10.28	9.07	7.13
hammer-human	3.23	2.19	2.90	1.75	2.50	3.72
hammer-cloned	9.66	0.56	7.11	14.52	11.65	8.24
relocate-human	0.12	0.03	0.05	0.07	0.05	0.04
relocate-cloned	1.20	−0.01	1.00	1.04	1.12	1.00

Table 11: D4RL means by domain and declared scope. Task means are averaged with equal weights; reward domains are not merged with OGBench success. Each result uses one training seed and 25 episodes per task. GFP's three MuJoCo tasks are medium-replay only, whereas FAC covers all nine MuJoCo source tasks. Adroit coverage is twelve tasks for GFP and eight for FAC. A broader all-measured mean and source-scope mean are both reported.

Inputs	Scope	Tasks	Actor	$\mathrm{FQE}_{.5}$	$\mathrm{BACI}_{.5}$	BACI_{1}
GFP	MuJoCo	3	33.21	35.25	32.23	32.72
GFP	AntMaze	6	67.33	71.33	68.67	72.67
GFP	Adroit	12	53.96	55.22	55.48	55.93
GFP	Source scope	18	58.42	60.59	59.87	61.51
GFP	All measured	21	54.81	56.97	55.92	57.39
FAC	MuJoCo	9	83.04	84.96	86.10	86.16
FAC	AntMaze	6	84.67	90.00	86.67	91.33
FAC	Adroit	8	25.38	25.97	25.97	25.96
FAC	Source scope	23	63.41	65.75	65.33	66.57

D.1 MATCHED DEPLOYMENT COMPARISON

Both Puzzle categories have the additional matched FQE-at-$p = 1$ control from the same 50K state. On GFP Puzzle-4×4, recursive selection has four more successes than FQE in 125 paired episodes. On FAC, both have 45 successes, but 16 episodes succeed exclusively for each method. Equal averages are not behavioral or population equivalence. The conditional intervals below do not describe training-seed variation.

Table 12: Matched $p = 1$ recurrence contrasts, BACI_1 minus FQE_1, in the raw-actor cohort. Intervals use 20,000 paired bootstrap resamples within each of five fixed goals, conditioning on the trained seed-zero weights; they do not cover training-seed variability and are not multiplicity-adjusted. The final two columns count episodes won exclusively by each method out of 125. A zero interval on saturated outcomes is not an equivalence certificate.

Backbone	Category	Difference (pp)	Conditional 95% CI	BACI-only	FQE-only
FAC	Puzzle-3×3	0.0	$[0.0, 0.0]$	0	0
FAC	Puzzle-4×4	0.0	$[-8.0, 8.0]$	16	16
GFP	Puzzle-3×3	-0.8	$[-2.4, 0.0]$	0	1
GFP	Puzzle-4×4	3.2	$[-4.0, 11.2]$	15	11

D.2 EVERY TARGET IN THE COMPONENT FACTORIAL

Table 13: Every target in the raw-GFP cost/radius factorial. Success percentages use the same 25 initializations per goal, one training seed, and matched 50K updates. Removing a constraint changes both training and deployment.

Goal	Both	No cost	No radius	Neither
antmaze-giant-navigate				
task1	0.0	0.0	0.0	0.0
task2	60.0	12.0	8.0	24.0
task3	56.0	56.0	68.0	48.0
task4	0.0	0.0	0.0	0.0
task5	28.0	36.0	16.0	36.0
antsoccer-navigate				
task1	84.0	92.0	88.0	76.0
task2	96.0	96.0	100.0	100.0
task3	64.0	44.0	68.0	52.0
task4	72.0	68.0	44.0	44.0
task5	36.0	52.0	16.0	44.0
cube-double-play				
task1	84.0	80.0	84.0	72.0
task2	52.0	56.0	60.0	48.0
task3	40.0	40.0	44.0	60.0
task4	8.0	4.0	0.0	0.0
task5	56.0	60.0	64.0	76.0

D.3 REWARD PREFIXES AND THE THEORETICAL OBJECTIVE

OGBench success and legacy D4RL normalized return are the primary benchmark metrics. The theoretical net return additionally subtracts accepted correction costs. Recorded trajectories are finite prefixes; Equation (12) includes a continuation boundary unless the MDP terminates. Observed reward extrema are not global reward bounds, and empirical TD errors are not value-error certificates. The following representative raw-actor records keep reward, cost, and net prefix separate.

Table 14: Logged discounted episode prefixes for the complete actor-reference comparisons. Each row averages five goals and 25 paired episodes per goal using the native backbone's discount. Net reward subtracts correction cost; higher is better. FQE $(0, 1)$ reuses the same trained critic as FQE $(0, .5)$. Values are not pooled across native recipes and do not include an estimated infinite-horizon boundary term.

Backbone	Category	Mode (t, p)	Reward	Cost	Net reward
FAC	Puzzle-3×3	FQE $(0, .5)$	-287.66	1.43	-289.10
FAC	Puzzle-3×3	FQE $(0, 1)$	-280.41	2.58	-283.00
FAC	Puzzle-3×3	BACI $(1, 1)$	-276.06	2.53	-278.58
FAC	Puzzle-3×3	BACI $(1, .5)$	-281.05	1.36	-282.41
FAC	Puzzle-3×3	Actor	-291.20	0.00	-291.20
FAC	Puzzle-4×4	FQE $(0, .5)$	-812.64	3.32	-815.95
FAC	Puzzle-4×4	FQE $(0, 1)$	-828.63	6.39	-835.02
FAC	Puzzle-4×4	BACI $(1, 1)$	-814.53	6.12	-820.65
FAC	Puzzle-4×4	BACI $(1, .5)$	-834.41	3.17	-837.57
FAC	Puzzle-4×4	Actor	-995.53	0.00	-995.53
GFP	Puzzle-3×3	FQE $(0, .5)$	-223.59	1.63	-225.22
GFP	Puzzle-3×3	FQE $(0, 1)$	-218.64	3.13	-221.77
GFP	Puzzle-3×3	BACI $(1, 1)$	-215.32	2.84	-218.16
GFP	Puzzle-3×3	BACI $(1, .5)$	-221.31	1.51	-222.82
GFP	Puzzle-3×3	Actor	-231.03	0.00	-231.03
GFP	Puzzle-4×4	FQE $(0, .5)$	-562.66	2.05	-564.72
GFP	Puzzle-4×4	FQE $(0, 1)$	-532.43	4.03	-536.46
GFP	Puzzle-4×4	BACI $(1, 1)$	-517.49	3.99	-521.48
GFP	Puzzle-4×4	BACI $(1, .5)$	-552.14	2.07	-554.21
GFP	Puzzle-4×4	Actor	-580.32	0.00	-580.32

D.4 Recursive Value Learning and Deployment Probability

Table 15 provides a finer comparison between the critic-training target and the deployment probability on the two Puzzle categories. The experiment uses three independent training seeds and 25 paired evaluation episodes per goal. For a fixed deployment probability, the FQE and BACI columns differ in whether the fitted critic evaluates only the actor ($t = 0$) or recursively includes future actor–flow selections ($t = 1$).

Table 15: Effect of recursive value learning and deployment probability on the Puzzle categories. Values are success rates (%) averaged over all five goals using three training seeds and 25 paired evaluation episodes per goal. FQE uses $t = 0$, whereas BACI uses $t = 1$. Within each method, changing p changes only the probability of executing an accepted flow proposal.

		$p = 0.5$		$p = 1$	
Actor–flow source	Category	FQE	BACI	FQE	BACI
GFP	puzzle-3×3-play	20.0	20.0	**20.0**	19.2
GFP	puzzle-4×4-play	24.0	25.6	26.4	**29.6**
FAC	puzzle-3×3-play	**100.0**	99.2	**100.0**	**100.0**
FAC	puzzle-4×4-play	35.2	32.8	**36.0**	**36.0**

The effect of recursive fitting varies across the two input pairs and Puzzle categories. At $p = 1$, BACI improves over FQE on GFP Puzzle-4×4 and matches FQE on FAC Puzzle-4×4, while their performance is similar on Puzzle-3×3. Changing p can also change the relative performance of the fitted critics. These comparisons show that recursive value learning and deployment probability are distinct factors in the resulting selection policy.