FLOW-TIME BELLMAN PROPAGATION FOR GENERATIVE POLICY IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow policies represent complex action distributions by transforming noise into actions through a sequence of generation steps. To learn useful behavior, these policies must incorporate value information into the generation process. Value-weighted flow matching favors high-value training actions, but leaves the target direction for each noise–action pair unchanged. We propose Flow-Time Bellman Propagation (FTBP), which learns value feedback for intermediate generation states and uses it to train the flow. A continuation potential predicts the critic value of the action obtained by completing the current flow. We train this potential with endpoint critic supervision and local bootstrap targets, avoiding a separate completed rollout for each intermediate target. The learned potential then guides training through successor evaluation (FTBP-S), target shaping (FTBP-T), or their combination (FTBP-C). During online fine-tuning, recorded generation traces provide additional supervision under the same policy objectives. Our analysis relates the two updates and characterizes value propagation for a matched finite-grid sampler. Experiments demonstrate that FTBP delivers strong and consistent improvements in both offline learning and online fine-tuning.

1 INTRODUCTION

Diffusion models and flow matching provide two expressive approaches to learning complex distributions from data (Ho et al., 2020; Lipman et al., 2023), motivating their use in trajectory planning and policy learning (Janner et al., 2022; Wang et al., 2023). These models generate behavior through a sequence of transformations, allowing them to represent diverse solutions to control problems. In reinforcement learning (RL), however, reproducing behavior is only part of the objective: the policy must also favor actions that lead to higher return. We study *generative policy improvement*, where value information guides the training of a generative model toward better behavior.

In offline RL, policy improvement must balance exploiting valuable behavior with controlling errors on actions poorly supported by the dataset (Kumar et al., 2020; Kostrikov et al., 2022). Flow policies provide a flexible representation for this setting and for subsequent online fine-tuning (Park et al., 2025b; Shin et al., 2026), but their expressivity alone does not specify how value information should guide training. Critic-optimized actor extraction and value-weighted imitation offer two ways to favor valuable actions (Park et al., 2025b; Tiofack et al., 2025). As illustrated by GFP in Figure 1, reweighting changes how strongly a noise–action pair contributes to training while retaining its conditional target direction. Improving individual generation steps therefore calls for feedback on how their updates affect the completed action.

The difficulty is that an intermediate state is still transformed by the remaining flow before becoming an action that the environment critic can evaluate. Q-Flow addresses this distinction by learning intermediate values from completed continuations (Doo et al., 2026), whereas internal TD methods learn values through diffusion transitions or reconstructed reverse-flow trajectories (Gao et al., 2025; Oberai et al., 2026). These approaches motivate learning feedback within generation, while leaving a concrete design problem for deterministic flow training: obtaining useful feedback from local transitions and incorporating it into velocity updates. We therefore ask: *how can locally bootstrapped value predictions be used to improve the training of a deterministic flow policy?*

We propose Flow-Time Bellman Propagation (FTBP), which learns reusable local value feedback for generative policy improvement. Our method trains a continuation potential to predict the critic

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.

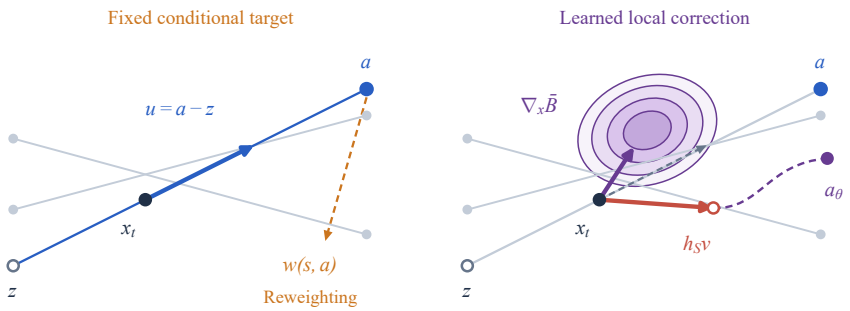


Figure 1: **Value feedback in flow training.** (a) GFP weights regression toward the unchanged target $u = a - z$ by $w(s, a)$. (b) FTBP adds local feedback; the dashed continuation ends at a_θ. Geometry is schematic; weighting also affects the learned flow. Bars denote target networks.

value of the action obtained by completing the current flow. At the endpoint, the critic provides direct supervision; at an intermediate state, the predicted value of a local successor supplies the training target. Through these local backups, endpoint information reaches the generation process without a separate completed rollout for each interior target. The learned potential supplies both value estimates and input gradients, allowing the same predictor to guide local flow updates alongside value-weighted imitation.

We use this feedback through successor evaluation (FTBP-S), target shaping (FTBP-T), or their combination (FTBP-C). During online fine-tuning, recorded generation traces provide additional supervision for the potential. We conduct experiments on OGBench (Park et al., 2025a) and D4RL (Fu et al., 2020), covering both offline learning and offline-to-online fine-tuning. The results show that FTBP consistently improves performance across both stages, highlighting the effectiveness of local value propagation within the flow.

2 Preliminaries

Offline and offline-to-online reinforcement learning. We consider a discounted Markov decision process $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, with state space $\mathcal{S}$, action space $\mathcal{A} \subseteq \mathbb{R}^{d_a}$, transition distribution P, reward function r, and discount factor $\gamma \in [0, 1)$. The objective is to learn a policy that maximizes expected discounted return. In offline RL, the policy is trained on a fixed dataset $\mathcal{D}$ of transitions (s, a, r, s', d), where $d = 1$ denotes termination (Levine et al., 2020). Offline-to-online RL adds a subsequent fine-tuning stage in which the policy collects new transitions through environment interaction. We write $\mathcal{R}$ for the replay distribution in either stage.

Given a bootstrap policy π, we estimate action values with a critic Q_ϕ, which is trained using the TD objective

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{\substack{(s,a,r,s',d)\sim\mathcal{R} \\ a'\sim\pi(\cdot|s')}} \left[\left(Q_\phi(s, a) - \mathrm{sg}[y] \right)^2 \right], \quad y = r + \gamma(1 - d) Q_{\bar{\phi}}(s', a'). \tag{1}$$

Here, a bar denotes target-network parameters and sg is the stop-gradient operator. We write the ensemble-aggregated critic as a scalar Q_ϕ for readability, with each head trained against the same target. The default bootstrap policy is the one-step actor described below.

Flow matching. A flow policy generates an action by transforming Gaussian noise with a state-conditioned velocity field v_θ (Lipman et al., 2023; Liu et al., 2023). For a fixed environment state s, generation follows

$$\frac{dx_t}{dt} = v_\theta(t, s, x_t), \qquad x_0 = z, \quad z \sim \mathcal{N}(0, I). \tag{2}$$

The variable $t \in [0, 1]$ indexes progress within the generation of a single action, while the environment state remains fixed throughout this process. Starting from noise z, we integrate the velocity

field to $t = 1$ and denote the resulting action by $a_\theta(s, z)$. Our implementation uses K Euler steps and projects the final output onto the normalized action space $\mathcal{A} = [-1, 1]^{d_a}$.

Flow matching trains the velocity field by regression, avoiding differentiation through the integration. Specifically, we sample a data action $(s, a) \sim \mathcal{R}$, noise $z \sim \mathcal{N}(0, I)$, and time $t \sim \mathcal{U}[0, 1]$. The interpolation $x_t = (1 - t)z + ta$ defines a conditional target velocity $u = a - z$. The velocity field is fitted to this target using

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{s,a,z,t}\left[\frac{w(s, a)}{d_a} \left\| v_\theta(t, s, x_t) - u \right\|_2^2\right]. \tag{3}$$

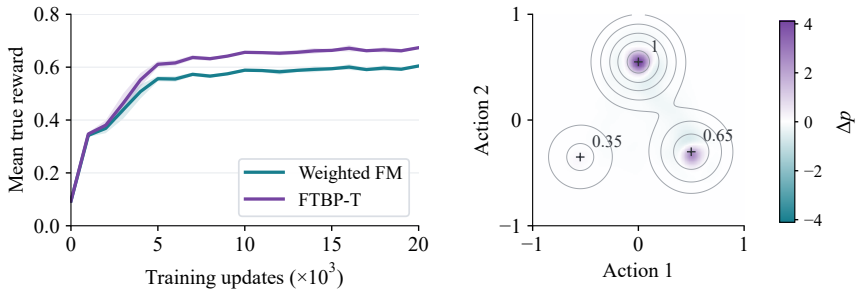
With $w(s, a) = 1$, this is the behavior-cloning objective used in FQL (Park et al., 2025b). GFP introduces nonnegative value-aware weights to emphasize useful actions (Tiofack et al., 2025). We adopt these weights and treat them as fixed during each flow update. The loss is averaged over the d_a action coordinates. Appendix C gives the full weight expression, including its dependence on sampled reference actions.

One-step policy extraction. To select actions without iterative integration, FQL trains a one-step actor by combining flow distillation with action-value maximization (Park et al., 2025b). Following this approach and GFP, we train an actor $f_\omega(s, z)$ with

$$\mathcal{L}_{\mathrm{actor}}(\omega) = \mathbb{E}_{s,z}\left[-c_Q Q_\phi(s, \mathrm{Proj}_{\mathcal{A}}\, f_\omega(s, z)) + \frac{\alpha}{d_a} \left\| f_\omega(s, z) - \mathrm{sg}[a_\theta(s, z)] \right\|_2^2\right]. \tag{4}$$

The critic term favors high-value actions, while the distillation term keeps the actor close to the action generated from the same noise by the flow. Here, $\mathrm{Proj}_{\mathcal{A}}$ projects onto the action space, $\alpha > 0$ controls distillation, and $c_Q > 0$ is a detached critic normalization factor. FTBP retains this extraction objective while modifying the training of the flow policy from which its distillation targets are obtained.

3 Motivation



(a) Reward (b) Density change

Figure 2: **Toy experiment.** (a) Mean reward across five paired seeds. (b) Final density difference, FTBP-T minus Weighted FM (purple: increase; teal: decrease).

Value weighting gives valuable data actions more influence during flow matching, but the target $u = a - z$ for each noise–action pair remains unchanged (Figure 1). Figure 2 asks what happens when local value feedback also shapes that target. In this single-state, two-dimensional toy problem, Weighted FM and FTBP-T share the data, value weights, initialization, and training budget; FTBP-T uses a learned continuation potential to shape the target. This comparison holds the preference over data actions fixed while changing the flow's local training signal. The complete protocol is in Appendix D.1.

In this setting, FTBP-T reaches higher reward during training and shifts its final action density toward the highest-reward mode (Figure 2). The difference under shared value weights points to the role of directional feedback: an intermediate flow state matters through the action produced by its remaining trajectory. Guiding that state therefore requires an estimate of its continuation value. FTBP learns this estimate from endpoint values and local successor predictions, then uses it to steer flow updates.

4 METHOD

In this section, we present Flow-Time Bellman Propagation (FTBP), which trains a deterministic flow policy using locally bootstrapped value feedback. As illustrated in Figure 3, the method has two components: a continuation potential that learns from endpoint critic values and local successor predictions, and a policy objective that uses this potential for successor evaluation, target shaping, or their combination. Both act on the same predictor, so the value information available at completed actions is carried to the intermediate states at which the flow is trained. We describe these components in turn and then explain how recorded generation traces provide additional supervision during online fine-tuning.

4.1 LEARNING INTERMEDIATE VALUE FEEDBACK

Since the feedback should reflect the action produced by the remaining generation steps, we define it in terms of the current flow's continuation. Let $F^{\theta}_{t:1}(s,x)$ denote the action obtained by continuing from (t,x) to the endpoint under a specified numerical sampler, with the flow and target critic held fixed. The corresponding continuation value is

$$B^{\theta}(t,s,x) = Q_{\bar{\phi}}\big(s, F^{\theta}_{t:1}(s,x)\big)\,. \tag{5}$$

Equation 5 assigns an intermediate state the critic value of its completed action, reducing to $B^{\theta}(1,s,a) = Q_{\bar{\phi}}(s,a)$ at the endpoint. At earlier times, it evaluates the continuation induced by the current flow, an interpretation also used in Q-Flow (Doo et al., 2026). To make this value directly available during local policy updates, we approximate it with a potential $B_{\psi}(t,s,x)$, which we train through endpoint supervision and local bootstrapping, thereby obtaining a predictor that can be queried at any intermediate state without completing the remaining generation.

The endpoint relation provides a direct way to supervise B_{ψ}: for an action a_b, we fit $B_{\psi}(1,s,a_b)$ to its target critic value. To train intermediate predictions, we use a local bootstrap target. For a step size $h \in (0,1)$ and $t \sim \mathcal{U}[0,1-h]$, we form a projected Euler successor and evaluate it with the target potential:

$$x^{+} = \mathrm{Proj}_{\mathcal{A}}(x + h\, v_{\theta}(t,s,x))\,, \qquad y_B = B_{\bar{\psi}}(t+h,s,x^{+}). \tag{6}$$

Fitting the prediction at (t,s,x) to y_B transfers the successor's predicted value to the current state, allowing intermediate supervision to use a single local transition instead of a completed continuation, thereby removing the rollout that Q-Flow requires for every sampled state. This is an internal TD backup, as used for diffusion policies in BDPO (Gao et al., 2025); here, however, the transition is deterministic and the value is anchored by a terminal critic with no running path cost. Combining the two sources of supervision gives

$$\begin{aligned}\mathcal{L}_B(\psi) = \mathbb{E}_{(s,a_b)\sim\nu_b}\left[\big(B_{\psi}(1,s,a_b) - \mathrm{sg}[Q_{\bar{\phi}}(s,a_b)]\big)^2\right] \\ + \mathbb{E}_{(s,t,x)\sim\nu}\left[\big(B_{\psi}(t,s,x) - \mathrm{sg}[y_B]\big)^2\right].\end{aligned} \tag{7}$$

The first term anchors endpoint predictions to the critic, while the second encourages consistency between predictions connected by a local flow transition. Both have unit weight in the full model; component ablations omit the corresponding term. Their targets are detached, so the objective updates only ψ. The two terms thereby train the potential across generation time from two sources that are both local: each requires only one velocity evaluation and one target-potential evaluation, rather than a completed suffix.

We draw endpoint actions from a mixture ν_b of replay actions, one-step actor samples, and flow-generated actions. Intermediate samples come from a mixture ν of clipped interpolation states and approximate flow states, as detailed in Appendix C. Each interior target uses a local velocity evaluation and a target-potential evaluation, avoiding a separate rollout of the remaining trajectory. The local transition also leaves the environment state unchanged, so its target contains no additional environmental reward or discount beyond the return already represented by the endpoint critic.

To characterize how these losses convey endpoint information, we consider a fixed flow and critic on a terminal-connected grid. Let $x_{j+1} = T_j(s,x_j)$ be its K numerical transitions and let $B^{\theta}_j(s,x_j) = Q(s,x_K)$ denote the resulting continuation value. The following proposition relates

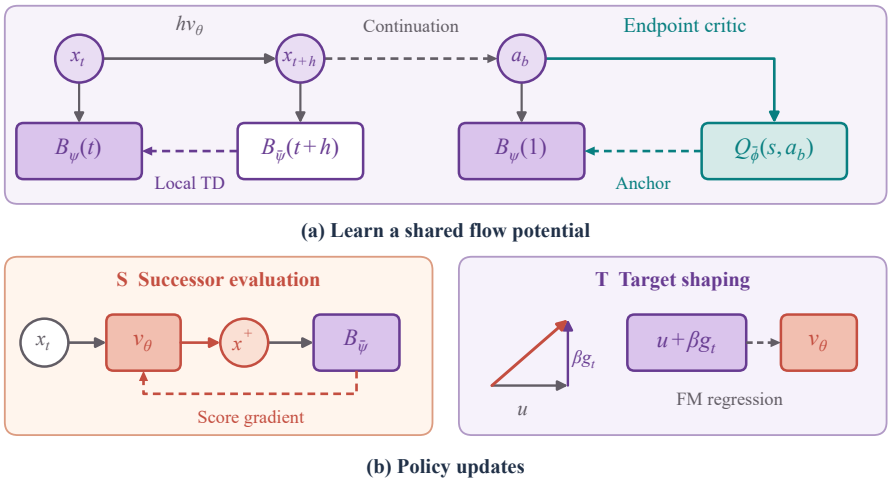


Figure 3: **Learning and applying local feedback.** (a) Local TD and endpoint critic supervision train the potential. (b) S optimizes successor value; T shapes the regression target. Dashed supervision arrows denote detached targets; the coral return path denotes the policy gradient.

prediction accuracy to the two errors controlled by Equation 7, provided the backup and continuation use the same transitions.

Proposition 1 (Fixed-grid consistency)**.** *Suppose functions* $\widehat{B}_0, \ldots, \widehat{B}_K$ *satisfy, along a continuation,*

$$\left|\widehat{B}_K(s, x_K) - Q(s, x_K)\right| \le \epsilon_b,$$
$$\left|\widehat{B}_j(s, x_j) - \widehat{B}_{j+1}(s, T_j(s, x_j))\right| \le \epsilon_j, \qquad j = k, \ldots, K-1.$$

Then

$$\left|\widehat{B}_k(s, x_k) - B_k^\theta(s, x_k)\right| \le \epsilon_b + \sum_{j=k}^{K-1} \epsilon_j. \tag{8}$$

In particular, zero boundary error and zero local residual at every state determine the continuation value uniquely.

Proposition 1 shows why endpoint anchoring and local consistency are used together: each local residual contributes to the error carried back from the endpoint. In the exact case, these conditions recover the value of the specified continuation without a completed-suffix target at every intermediate state. The proof is given in Appendix A.2; the distinction between this matched-grid result and our continuous-time implementation is discussed in Appendix C.1.

4.2 Improving the flow with local feedback

The potential is queried during flow updates as follows. We incorporate it into the weighted regression objective of Equation 3. During a flow update, the target-potential parameters $\bar{\psi}$ are held fixed and only the input derivative of the potential passes the value signal to the velocity field, thereby leaving the value estimator itself unaffected by the policy update. Below, we write $v = v_\theta(t, s, x_t)$ for the velocity at the sampled intermediate state.

FTBP-S: successor evaluation. The first approach evaluates the current velocity through the successor it proposes, $x_S^+ = \mathrm{Proj}_{\mathcal{A}}(x_t + h_S v)$, whose potential is queried at $t_S^+ = \min(t + h_S, 1)$. We add $-c_B B_{\bar{\psi}}(t_S^+, s, x_S^+)$ to the regression objective, where $c_B > 0$ is a detached normalization factor, and differentiate through the successor's dependence on v. This encourages moves with higher predicted continuation value while retaining the data-based regression signal, thereby steering the velocity toward successors that the potential rates more highly.

FTBP-T: target shaping. The second approach modifies the conditional target using $g_t = \mathcal{C}(\nabla_x B_{\bar{\psi}}(t, s, x_t))$, where $\mathcal{C}$ clips the gradient norm before it is used as guidance. Replacing u with $u + \beta g_t$ for $\beta > 0$ retains the data-based direction while adding a component that points toward increasing predicted continuation value. We detach the resulting target before regression, treating g_t as fixed supervision for the velocity field, thereby avoiding differentiation through the construction of the input gradient.

FTBP-C: combined feedback. Target shaping specifies a desired direction at the current state, whereas successor evaluation scores the model's proposed move. Since these need not agree for a finite step, we also consider using both. The three variants share the objective

$$\mathcal{L}_{\text{flow}}(\theta; \beta, \lambda) = \mathbb{E}_{s,a,z,t}\Bigg[\frac{w(s,a)}{d_a}\,\|v - \text{sg}[u + \beta g_t]\|_2^2 - \lambda c_B B_{\bar{\psi}}(t_S^+, s, x_S^+)\Bigg]. \tag{9}$$

FTBP-S uses $\beta = 0$ and $\lambda > 0$, FTBP-T uses $\beta > 0$ and $\lambda = 0$, and FTBP-C uses both positive, while setting both coefficients to zero recovers the original weighted FM loss. The successor term passes gradients into the velocity field through the proposed move, whereas the regression term treats the shaped target as fixed. Both provide local updates without requiring backpropagation through a complete generation trajectory.

We compare the per-sample velocity gradients to characterize the two feedback mechanisms at the level of the local policy objective, independently of parameter sharing in the velocity network.

Proposition 2 (First-order relation between S and T)**.** *Fix a training sample and its detached quantities u, w, c_B, and write $b(t,x) = B_{\bar{\psi}}(t,s,x)$ and $g = \nabla_x b(t,x)$. Let ℓ_S and ℓ_T be the per-sample losses in Equation 9 with $(\beta,\lambda) = (0,\lambda)$ and $(\beta, 0)$, respectively. If b is twice continuously differentiable, v is bounded, $t+h_S \le 1$, and projection and gradient clipping are inactive in a neighborhood of the update, then, as $h_S \to 0$,*

$$\nabla_v \ell_S - \nabla_v \ell_T = \left(\frac{2w\beta}{d_a} - \lambda c_B h_S\right) g + O(h_S^2). \tag{10}$$

Here λ, c_B are held fixed, and the remainder is measured in Euclidean norm.

Proposition 2 identifies a common first-order effect: both objectives add a preference in the potential-gradient direction to the FM update. Their strengths match for a given sample when $2w\beta/d_a = \lambda c_B h_S$, although sample-dependent weights generally prevent a single global coefficient match. For finite steps, S evaluates the gradient at the proposed successor, whereas T shapes the target at the current state. This motivates comparing both interfaces and their combination; Appendix A.5 provides the proof and discusses clipping.

4.3 Additional supervision during online fine-tuning

During online fine-tuning, we store sparse traces $\tau = \{(t_j, \tilde{x}_j)\}_{j=1}^m$ from flow-generated actions, where each $\tilde{x}_j$ is a clipped intermediate snapshot. The environmental TD target y from Equation 1 supervises these points using the resulting transition:

$$\mathcal{L}_{\text{trace}}(\psi) = \mathbb{E}_{s,\tau,y}\left[\frac{1}{m}\sum_{j=1}^{m}\left(B_\psi(t_j, s, \tilde{x}_j) - \text{sg}[y]\right)^2\right]. \tag{11}$$

The recorded points share y because they produced the same executed action. Its reward and next-state bootstrap add interaction-based supervision to the local potential loss. Older traces can differ from the current flow, so their targets are auxiliary. FINO instead targets exploration during flow training and sampling (Shin et al., 2026). Endpoint and adjacent-trace losses, along with numerical details, appear in Appendix C.

Table 1: **Offline OGBench results** under the FQL setting (Park et al., 2025b), after 1M updates. Results are mean$_{\pm \mathrm{SD}}$ across more than 10 seeds; bold marks the highest mean.

Environment (5 tasks each)	IQL	ReBRAC	FQL	IFQL	GFP	Q-Flow	FTBP-S	FTBP-T	FTBP-C
antmaze-large-navigate	$53_{\pm 3}$	$81_{\pm 5}$	$79_{\pm 3}$	$28_{\pm 5}$	$94_{\pm 2}$	$89_{\pm 5}$	$93_{\pm 2}$	$95_{\pm 2}$	$\mathbf{96}_{\pm \mathbf{1}}$
antmaze-giant-navigate	$4_{\pm 1}$	$26_{\pm 8}$	$9_{\pm 6}$	$3_{\pm 2}$	$28_{\pm 9}$	$40_{\pm 4}$	$47_{\pm 6}$	$33_{\pm 7}$	$\mathbf{48}_{\pm \mathbf{5}}$
humanoidmaze-medium-navigate	$33_{\pm 2}$	$22_{\pm 8}$	$58_{\pm 5}$	$60_{\pm 14}$	$72_{\pm 3}$	$\mathbf{83}_{\pm \mathbf{4}}$	$72_{\pm 3}$	$73_{\pm 2}$	$73_{\pm 2}$
humanoidmaze-large-navigate	$2_{\pm 1}$	$2_{\pm 1}$	$4_{\pm 2}$	$11_{\pm 2}$	$18_{\pm 10}$	$8_{\pm 2}$	$17_{\pm 7}$	$11_{\pm 7}$	$\mathbf{20}_{\pm \mathbf{8}}$
antsoccer-arena-navigate	$8_{\pm 2}$	$0_{\pm 0}$	$60_{\pm 2}$	$33_{\pm 6}$	$58_{\pm 2}$	$56_{\pm 4}$	$63_{\pm 4}$	$64_{\pm 6}$	$\mathbf{65}_{\pm \mathbf{4}}$
cube-double-play	$7_{\pm 1}$	$12_{\pm 1}$	$29_{\pm 2}$	$14_{\pm 3}$	$47_{\pm 2}$	$36_{\pm 3}$	$43_{\pm 3}$	$44_{\pm 4}$	$\mathbf{51}_{\pm \mathbf{4}}$
puzzle-4×4-play	$7_{\pm 1}$	$14_{\pm 1}$	$17_{\pm 2}$	$25_{\pm 5}$	$26_{\pm 2}$	$29_{\pm 2}$	$27_{\pm 4}$	$27_{\pm 3}$	$\mathbf{29}_{\pm \mathbf{3}}$
Average	16.3	22.4	36.6	24.9	49.0	48.7	51.7	49.7	**54.6**

5 EXPERIMENTS

In this section, we ask four questions that follow directly from the method design. Section 5.1 asks whether local value feedback improves offline learning and subsequent online adaptation. Section 5.2 asks whether the location of feedback within the flow update changes the learning trajectory. Section 5.3 separates the contribution of local TD supervision from that of online generation traces. Finally, Section 5.4 tests sensitivity to the two main feedback parameters and reports the resulting step-time cost. We keep the task families and training budgets fixed across these comparisons so that each experiment isolates one part of the proposed mechanism.

Environments. We use OGBench (Park et al., 2025a) as the primary testbed because its single-task formulation turns each goal into an explicit task while covering long-horizon navigation, humanoid control, contact-rich manipulation, and sparse-reward puzzles. These task families stress the same question from different angles: whether feedback learned inside the flow remains useful across action dimensions, dynamics, and goal structure. The main comparison averages five goals in each of seven `navigate` or `play` environments, with equal weight across environment families. Stitch and noisy variants, together with D4RL AntMaze (Fu et al., 2020), provide a cross-benchmark check in Appendix E; Appendix D lists the task identifiers.

Baselines. We choose baselines to identify where value enters the policy and which part of the feedback pipeline accounts for an improvement. IQL and ReBRAC provide non-flow references based on in-sample learning and behavior regularization (Kostrikov et al., 2022; Tarasov et al., 2023). FQL is the one-step flow backbone, while IFQL tests whether selecting among multiple generated candidates is sufficient (Park et al., 2025b). GFP is the closest control because it changes the weight of each flow-matching example while leaving its target direction unchanged (Tiofack et al., 2025). Q-Flow supplies a flow-level comparison that learns intermediate values from completed continuations (Doo et al., 2026). For online fine-tuning, RLPD measures adaptation without offline pre-training and FINO represents exploration-oriented flow training and action selection (Ball et al., 2023; Shin et al., 2026). Together, these baselines distinguish reweighting, post-hoc action selection, continuation guidance, and exploration from the local feedback studied here.

Evaluation. We pretrain for 1M gradient updates and fine-tune for 500K environment interactions; RLPD uses the same online budget but starts directly from the offline dataset. Each environment entry averages five goals, and the Average row weights the seven environment families equally. To give every baseline its strongest published reference point, we place the best result reported in the corresponding baseline paper under the matching setting in the main tables. We also reimplement the baselines under our protocol; the complete reproduced results are reported in Tables 4–7. This separates the strongest published comparison from the reproducibility check without mixing their evaluation protocols. Published offline SDs are copied as reported, while FINO's 95% confidence intervals over 10 seeds are converted to approximate SDs using $\mathrm{CI}\sqrt{10}/1.96$. Appendix D gives the checkpoint and episode procedures.

Table 2: **Online OGBench results** under the FQL setting (Park et al., 2025b), after 500K interactions. Results are mean$_{\pm \mathrm{SD}}$ across more than 10 seeds; bold marks the highest mean.

Environment (5 tasks each)	ReBRAC	RLPD	FQL	IFQL	FINO	Q-Flow	FTBP-S	FTBP-T	FTBP-C
antmaze-large-navigate	$99_{\pm 0}$	$80_{\pm 11}$	$92_{\pm 2}$	$72_{\pm 10}$	$99_{\pm 0}$	$96_{\pm 1}$	$99_{\pm 1}$	$\mathbf{99_{\pm 0}}$	$99_{\pm 1}$
antmaze-giant-navigate	$\mathbf{96_{\pm 6}}$	$47_{\pm 15}$	$71_{\pm 8}$	$0_{\pm 0}$	$79_{\pm 0}$	$85_{\pm 9}$	$88_{\pm 2}$	$85_{\pm 3}$	$92_{\pm 2}$
humanoidmaze-medium-navigate	$3_{\pm 5}$	$1_{\pm 2}$	$61_{\pm 3}$	$70_{\pm 8}$	$\mathbf{97_{\pm 2}}$	$93_{\pm 4}$	$78_{\pm 1}$	$78_{\pm 1}$	$79_{\pm 1}$
humanoidmaze-large-navigate	$1_{\pm 0}$	$0_{\pm 0}$	$10_{\pm 5}$	$10_{\pm 3}$	$\mathbf{33_{\pm 11}}$	$10_{\pm 2}$	$27_{\pm 11}$	$25_{\pm 14}$	$32_{\pm 13}$
antsoccer-arena-navigate	$0_{\pm 0}$	$2_{\pm 2}$	$74_{\pm 6}$	$35_{\pm 8}$	$77_{\pm 8}$	$82_{\pm 3}$	$91_{\pm 2}$	$91_{\pm 3}$	$\mathbf{91_{\pm 2}}$
cube-double-play	$28_{\pm 3}$	$2_{\pm 5}$	$73_{\pm 3}$	$40_{\pm 3}$	$\mathbf{79_{\pm 3}}$	$74_{\pm 3}$	$79_{\pm 3}$	$78_{\pm 1}$	$78_{\pm 2}$
puzzle-4×4-play	$29_{\pm 8}$	$58_{\pm 18}$	$45_{\pm 13}$	$42_{\pm 6}$	$56_{\pm 8}$	$40_{\pm 0}$	$52_{\pm 7}$	$66_{\pm 8}$	$\mathbf{68_{\pm 8}}$
Average	36.6	27.1	60.9	38.4	74.3	68.6	73.4	74.5	**77.0**

5.1 Offline and online performance

Tables 1 and 2 test the same question across the two training phases: whether local value feedback improves the policy and remains useful during online adaptation. The offline comparison separates FTBP from value-weighted flow matching and completed-continuation guidance, while the online comparison tests whether the learned policy can exploit new interaction. FTBP-C gives the most consistent aggregate behavior, with FINO remaining competitive on exploration-sensitive tasks. This complementary pattern indicates that local feedback improves value propagation, while exploration still matters when coverage limits adaptation.

5.2 Local feedback ablations

Figure 4 compares GFP with FTBP-S/T/C during offline training on two navigation goals selected to expose local guidance before the final checkpoint. The trajectories show how each interface changes learning, rather than only its endpoint.

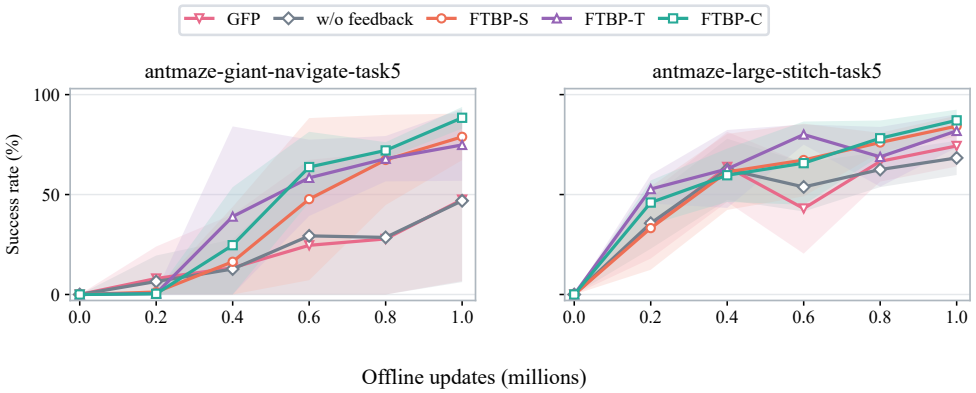


Figure 4: **Local feedback during offline training.** Two AntMaze goals; bands show standard deviation across runs.

The curves show that local feedback changes the learning trajectory. Successor evaluation and target shaping provide distinct routes for injecting continuation value, and their combination is the most consistent across the two goals. This supports the objective-level view that the two signals complement one another.

5.3 Local TD and online trace supervision

Figure 5 isolates the two supervision sources with matched five-goal averages from humanoidmaze-medium-navigate and cube-double-play. The offline panels remove local TD targets, whereas the online panels remove trace supervision after interaction begins.

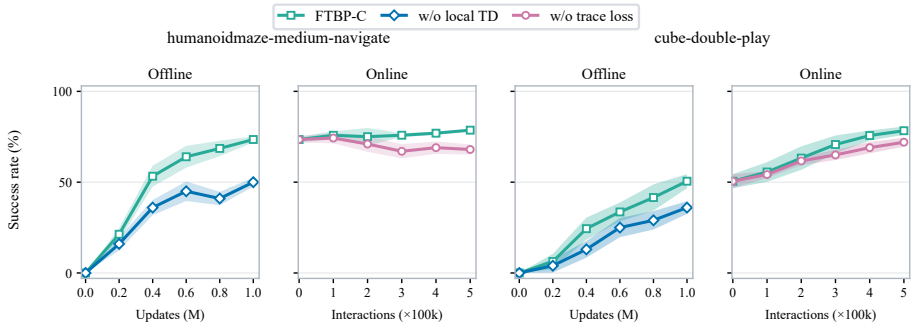


Figure 5: **Local TD and trace ablations.** Five-goal task families; bands show standard deviation.

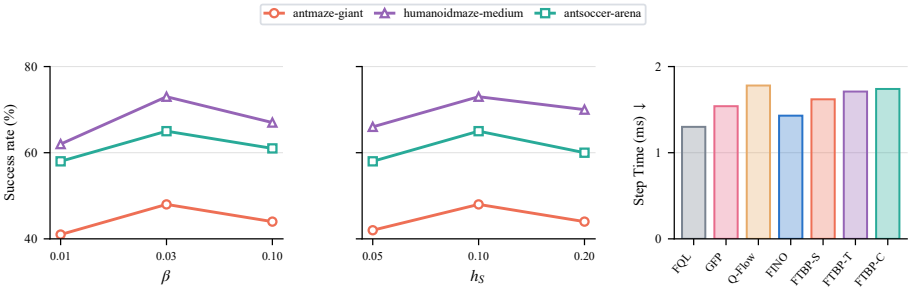


Figure 6: **Sensitivity and training cost.** Parameter sweeps and matched step time; lines show offline means for three task families, and lower time is better.

The paired curves reveal an asymmetric role for the two losses. Local TD targets are needed to establish a useful continuation potential during offline training, while trace loss becomes useful once the policy collects new transitions. Removing either source weakens the corresponding stage, showing that online traces extend the offline value representation rather than replace its local bootstrap.

5.4 Parameter sensitivity and efficiency

We next test whether the default feedback parameters are stable choices. Figure 6 varies β and h_S one at a time while keeping checkpoints, seeds, budgets, and all other settings fixed. The centered default is the most reliable setting across the three task families, and the response remains smooth on either side of it. The step-time panel shows that the feedback variants stay within the same computational range as the other flow baselines, so the performance gains do not require a distinct inference-time cost profile. Together, these results suggest that FTBP provides a robust and practical way to incorporate local value feedback without requiring delicate hyperparameter tuning or substantial computational overhead.

6 Conclusion

We presented FTBP, a framework for improving flow policies with locally learned value feedback. FTBP learns a continuation potential from endpoint critic supervision and local bootstrap targets, then exposes that potential to the flow through successor evaluation, target shaping, or their combination. Because the same predictor can be updated with recorded generation traces during online fine-tuning, the method connects endpoint values to the intermediate states that determine each flow update. Experiments on OGBench show that combined feedback is the most reliable configuration across offline learning and online adaptation, while the ablations show that local TD supervision is central to offline propagation and trace supervision extends the potential during interaction. The sensitivity and efficiency results further indicate that the default feedback settings provide a stable accuracy–cost trade-off. Future work should study how to maintain calibrated local feedback as the policy and its numerical sampler evolve.

Ethics Statement

This work studies general-purpose reinforcement-learning algorithms in simulated control domains and does not involve human subjects or private data. Generative policies can be deployed in safety-critical systems; the experiments in this paper do not establish real-world safety, and such deployment would require domain-specific validation and safeguards.

Reproducibility Statement

The main text specifies the learning objectives, the evaluation protocol, and the measured training cost, and Appendix A provides the fixed-flow consistency proof. Appendix C gives the architectures and the hyperparameter configuration shared across variants and tasks, Appendix D describes dataset construction, checkpointing, and evaluation, and Appendix E reports per-task results with seed-level statistics. An anonymized implementation and the commands for reproducing all tables and figures are included in the supplementary material.

AI Use Statement

Large language models were used solely for writing refinement and language polishing; they were not used for any other part of this work.

References

Bhavya Agrawalla, Michal Nauman, Khush Agrawal, and Aviral Kumar. floq: Training critics via flow-matching for scaling compute in value-based reinforcement learning. *arXiv preprint arXiv:2509.06863*, 2025.

Bhavya Agrawalla, Michal Nauman, and Aviral Kumar. What does flow matching bring to TD learning? *arXiv preprint arXiv:2603.04333*, 2026.

Marvin Alles, Nutan Chen, Patrick van der Smagt, and Botond Cseke. FlowQ: Energy-guided flow policies for offline reinforcement learning. *arXiv preprint arXiv:2505.14139*, 2025.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 2023.

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.

Deshu Chen, Yuchen Liu, Zhijian Zhou, Chao Qu, and Yuan Qi. Unleashing flow policies with distributional critics. *arXiv preprint arXiv:2509.23087*, 2025.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

Yonghoon Dong, Kyungmin Lee, Changyeon Kim, Jaehyuk Kim, and Jinwoo Shin. Trust region Q adjoint matching. *arXiv preprint arXiv:2605.27079*, 2026. URL https://arxiv.org/abs/2605.27079.

JaeHyeok Doo, Byeongguk Jeon, Seonghyeon Ye, Kimin Lee, and Minjoon Seo. Q-flow: Stable and expressive reinforcement learning with flow-based policy. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Nicolas Espinosa-Dice, Kiante Brantley, and Wen Sun. Expressive value learning for scalable offline reinforcement learning. *arXiv preprint arXiv:2510.08218*, 2025.

Linjiajie Fang, Ruoxue Liu, Jing Zhang, Wenjia Wang, and Bing-Yi Jing. Diffusion actor-critic: Formulating constrained policy iteration as diffusion noise regression for offline reinforcement learning. *arXiv preprint arXiv:2405.20555*, 2024.

Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. *arXiv preprint arXiv:2502.02150*, 2025.

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.

Chen-Xiao Gao, Chenyang Wu, Mingjun Cao, Chenjun Xiao, Yang Yu, and Zongzhang Zhang. Behavior-regularized diffusion policy optimization for offline reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18630–18657, 2025. URL https://proceedings.mlr.press/v267/gao25q.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 9902–9915. PMLR, 2022. URL https://proceedings.mlr.press/v162/janner22a.html.

Fairoz Nower Khan, Nabuat Zaman Nahim, Ruiquan Huang, Haibo Yang, and Peizhong Ju. Flow matching for offline reinforcement learning with discrete actions. *arXiv preprint arXiv:2602.06138*, 2026.

Prajwal Koirala and Cody Fleming. Flow-based single-step completion for efficient and expressive policy learning. *arXiv preprint arXiv:2506.21427*, 2025.

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

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Boshu Lei, Kostas Daniilidis, and Antonio Loquercio. Reinforcement learning for flow-matching policies with density transport. *arXiv preprint arXiv:2606.08602*, 2026.

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.

Mingxuan Li, Junzhe Zhang, and Elias Bareinboim. Causal flow Q-learning for robust offline reinforcement learning. *arXiv preprint arXiv:2602.02847*, 2026a.

Qiyang Li and Sergey Levine. Q-learning with adjoint matching. In *International Conference on Learning Representations*, 2026.

Zeyang Li, Sunbochen Tang, and Navid Azizan. Reverse flow matching: A unified framework for online reinforcement learning with diffusion and flow policies. *arXiv preprint arXiv:2601.08136*, 2026b.

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.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online reinforcement learning. *arXiv preprint arXiv:2505.05470*, 2025a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Zhen Liu, Tim Z. Xiao, Carles Domingo-Enrich, Weiyang Liu, and Dinghuai Zhang. Value gradient guidance for flow matching alignment. In *Advances in Neural Information Processing Systems*, 2025b.

Lei Lv, Yunfei Li, Yu Luo, Fuchun Sun, Tao Kong, Jiafeng Xu, and Xiao Ma. Flow-based policy for online reinforcement learning. *arXiv preprint arXiv:2506.12811*, 2025.

Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. *arXiv preprint arXiv:2502.00361*, 2025.

Liyuan Mao, Haoran Xu, Xianyuan Zhan, Weinan Zhang, and Amy Zhang. Diffusion-dice: In-sample diffusion guidance for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

Yunpeng Mei, Jiakai He, Hongjie Cao, Chenyu Wang, Xiaowen Zhu, Yihan Zhou, Jiamin Wang, Chenbo Xin, Peng Cheng, Yuxuan Yang, Yijie Wang, Xinhu Zheng, Gao Huang, Jie Chen, and Gang Wang. Potential-guided flow matching for vision-language-action policy improvement. *arXiv preprint arXiv:2606.04968*, 2026.

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 *Advances in Neural Information Processing Systems*, 2023.

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.

Thanh Nguyen, Tri Ton, Hongbin Choe, Tung M. Luu, and Chang D. Yoo. Fast and highly expressive policy learning for offline reinforcement learning via bootstrapped flow Q-learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Aditya Oberai, Seohong Park, and Sergey Levine. Reversal Q-learning. *arXiv preprint arXiv:2606.17551*, 2026. URL `https://arxiv.org/abs/2606.17551`.

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

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025b.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Xiaotian Qiu, Lukai Chen, Jinhao Li, Qi Sun, Cheng Zhuo, and Guohao Dai. ScoRe-Flow: Complete distributional control via score-based reinforcement learning for flow matching. *arXiv preprint arXiv:2604.10962*, 2026.

Allen Z. Ren, Justin Lidard, Lars L. Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, 2025.

Yifan Ruan, Chenyang Cao, Andreas Burger, Ali Pesaranghader, Kaveh Kamali, Jaehong Kim, Nandita Vijaykumar, Alan Aspuru-Guzik, Igor Gilitschenski, and Nicholas Rhinehart. QPILOTS: Efficient test-time Q-steering for flow policies. *arXiv preprint arXiv:2606.14801*, 2026.

Yash Savani, Branislav Kveton, Yuchen Liu, Yilin Wang, Jing Shi, Subhojyoti Mukherjee, Nikos Vlassis, and Krishna Kumar Singh. Stepwise credit assignment for GRPO on flow-matching models. *arXiv preprint arXiv:2603.28718*, 2026.

Kexin Shi, Junyao Shi, Poorvi Hebbar, Zhuolun Zhao, Tarun Amarnath, Yifan Su, Shikhar Bahl, and Deepak Pathak. FlowDPG: Deterministic policy gradient on flow matching policies for real-world manipulation. *arXiv preprint arXiv:2606.22303*, 2026.

Yongjae Shin, Jongseong Chae, Jongeui Park, and Youngchul Sung. Flow matching with injected noise for offline-to-online reinforcement learning. In *International Conference on Learning Representations*, 2026.

Andreas Sochopoulos, Esmeralda S. Whitammer, Nikolaos Tsagkas, João Moura, Michael Gienger, and Sethu Vijayakumar. Optimal transport Q-learning for flow policy steering and acceleration. *arXiv preprint arXiv:2607.06262*, 2026.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.09836.

Manan Tayal and Mumuksh Tayal. Epigraph-guided flow matching for safe and performant offline reinforcement learning. *arXiv preprint arXiv:2602.08054*, 2026.

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. *arXiv preprint arXiv:2512.03973*, 2025.

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

Ziqian Wang, Yitian Liu, Xingjian Mao, Minqian Wang, and Yao Mu. Q-VGM: Q-value-gradient matching for offline-to-online reinforcement learning of flow-matching VLA. *arXiv preprint arXiv:2606.08015*, 2026.

Liuhaichen Yang, Zhuang Jiang, Chenchao Sheng, and Zezhi Tang. Guided action flow: Q-guided inference for flow-matching vision-language-action policies. *arXiv preprint arXiv:2607.02092*, 2026a.

Ningyuan Yang, Jiaxuan Gao, Feng Gao, Yi Wu, and Chao Yu. Fine-tuning diffusion policies with backpropagation through diffusion timesteps. *arXiv preprint arXiv:2505.10482*, 2025.

Rushuai Yang, Zhuo Han, Houlin Li, Hecheng Wang, Zhichao Wu, Rui Zhang, Zhaowei Zhang, Zihong Chen, Xiaohan Yan, Chiming Liu, Yi Chen, Wei Shan, and Maoqing Yao. VINE: Taming generative control policies for reinforcement learning. *arXiv preprint arXiv:2607.10369*, 2026b.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *International Conference on Learning Representations*, 2025a.

Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. ReinFlow: Fine-tuning flow matching policy with online reinforcement learning. *arXiv preprint arXiv:2505.22094*, 2025b.

Hai Zhong, Zhuoran Li, Xun Wang, and Longbo Huang. Reparameterization flow policy optimization. *arXiv preprint arXiv:2602.03501*, 2026.

Shan Zhong, Shutong Ding, He Diao, Xiangyu Wang, Kah Chan Teh, and Bei Peng. FlowCritic: Bridging value estimation with flow matching in reinforcement learning. *arXiv preprint arXiv:2510.22686*, 2025.

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.

Christos Ziakas, Alessandra Russo, and Avishek Joey Bose. Aligning flow map policies with optimal Q-guidance. *arXiv preprint arXiv:2605.12416*, 2026.

Guowei Zou, Haitao Wang, Hejun Wu, Yukun Qian, Yuhang Wang, and Weibing Li. One step is enough: Dispersive MeanFlow policy optimization. *arXiv preprint arXiv:2601.20701*, 2026.

A THEORETICAL DETAILS

We begin with the continuation value of a fixed flow. A terminal-connected sampler yields an exact local identity, from which we bound prediction and gradient errors. We then prove the S–T relation in Proposition 2 and bound the gain from an approximate value gradient. Throughout the continuation analysis, the environment state, flow, critic, and sampler are fixed. The policy-update comparison varies the proposed velocity while holding the potential fixed.

A.1 CONTINUATION VALUES AND TRANSPORT ALONG A FIXED FLOW

Fix a state s and a grid $0 = t_0 < \cdots < t_K = 1$. Let $T_j(s, \cdot)$ be the deterministic numerical transition from t_j to t_{j+1}, including any projection prescribed by the reference sampler. For each starting point $x_k = x$, define

$$x_{j+1} = T_j(s, x_j), \qquad B_k^\theta(s, x) = Q(s, x_K), \qquad B_K^\theta(s, a) = Q(s, a). \tag{12}$$

Continuing from x_k or its successor x_{k+1} reaches the same endpoint. Thus,

$$B_k^\theta(s, x) = B_{k+1}^\theta(s, T_k(s, x)). \tag{13}$$

Equation 13 motivates the local backup target. An internal transition leaves the environment state unchanged and collects no reward. Because Q already incorporates environmental discounting, the identity propagates value with unit discount.

In continuous time, the flow transports the continuation value along its trajectories. Suppose v_θ is continuously differentiable on a region containing the trajectories and generates a unique flow map $\Phi_{t:u}^\theta$. Let the terminal payoff q_s be continuously differentiable on the corresponding endpoint region. For an unprojected flow, $q_s(x) = Q(s, x)$; endpoint projection can be included in $q_s(x) = Q(s, \mathrm{Proj}_{\mathcal{A}}\, x)$ wherever this composite is differentiable. Define

$$B^\theta(t, s, x) = q_s\big(\Phi_{t:1}^\theta(s, x)\big).$$

By the composition property of the flow map, $B^\theta(u, s, \Phi_{t:u}^\theta(s, x)) = B^\theta(t, s, x)$. Differentiating with respect to u gives the transport equation

$$\partial_t B^\theta(t, s, x) + \nabla_x B^\theta(t, s, x)^\top v_\theta(t, s, x) = 0, \qquad B^\theta(1, s, x) = q_s(x). \tag{14}$$

Conversely, a continuously differentiable solution of Equation 14 is constant along each characteristic. Its terminal condition therefore identifies it with the continuation value on every trajectory that reaches $t = 1$ within the stated region.

Equation 14 also explains the local backup. For a twice continuously differentiable candidate b, a bounded velocity, and an unprojected step with $t + h \le 1$,

$$b(t + h, s, x + h v_\theta(t, s, x)) - b(t, s, x) = h\big(\partial_t b + \nabla_x b^\top v_\theta\big)(t, s, x) + O(h^2).$$

The finite-step loss fits neighboring predictions along this transport direction. The consistency proof below instead uses the exact grid identity.

A.2 FIXED-GRID CONSISTENCY

To prove Proposition 1, fix the state, critic, and transitions in Equation 12. Write the boundary error and local residuals of a candidate potential as

$$e_K(x) = \widehat{B}_K(s, x) - Q(s, x), \qquad r_j(x) = \widehat{B}_j(s, x) - \widehat{B}_{j+1}(s, T_j(s, x)).$$

Proof. Let $e_j(x) = \widehat{B}_j(s, x) - B_j^\theta(s, x)$. Substituting Equation 13 into the definition of r_j yields

$$e_j(x_j) = r_j(x_j) + e_{j+1}(x_{j+1}).$$

Substituting from $j = k$ through $j = K - 1$ gives the exact decomposition

$$e_k(x_k) = e_K(x_K) + \sum_{j=k}^{K-1} r_j(x_j). \tag{15}$$

Applying the triangle inequality and the assumed residual bounds proves $|e_k(x_k)| \le \epsilon_b + \sum_{j=k}^{K-1} \epsilon_j$. For uniqueness, suppose the boundary and residuals vanish at every state. Any two solutions then agree at K. If they agree at $j+1$, Equation 13 makes them agree at j for every state. Backward induction establishes equality on the entire grid.

Each residual contributes once along the continuation. With uniform bounds $\epsilon_j \le \epsilon_{\mathrm{td}}$, the error is at most $\epsilon_b + (K-k)\epsilon_{\mathrm{td}}$. Thus, an exact backward sweep recovers the continuation value on a matched grid. Training, however, fits sampled residuals with a lagged target; the next result accounts for both.

A.3 Prediction error under sampled supervision

The pointwise bounds in Proposition 1 do not directly describe training under sampled distributions. We therefore bound prediction error on a queried continuation distribution using population losses and coverage. Let ρ_k be a distribution over (s, x_k), and let ρ_j be its pushforward through the fixed transitions to time t_j. Let ν_j supervise the local residual at that time and ν_K supervise the boundary.

Proposition 3 (Prediction error with coverage and target lag)**.** *Let $\widehat{B}_j$ and $\overline{B}_j$ be candidate and target potentials. Define $\delta_j(s,x) = \widehat{B}_j(s,x) - \overline{B}_{j+1}(s, T_j(s,x))$. Suppose ρ_j has a density relative to ν_j bounded by $C_j < \infty$ for every $j = k, \dots, K$. Assume*

$$\mathbb{E}_{\nu_j}[\delta_j^2] \le \ell_j \quad (j = k, \dots, K-1), \qquad \mathbb{E}_{\nu_K}[(\widehat{B}_K - Q)^2] \le \ell_b.$$

If $\|\overline{B}_{j+1} - \widehat{B}_{j+1}\|_{L^2(\rho_{j+1})} \le d_{j+1}$ for every $j = k, \dots, K-1$, then

$$\|\widehat{B}_k - B_k^\theta\|_{L^2(\rho_k)} \le \sqrt{C_K \ell_b} + \sum_{j=k}^{K-1} \left(\sqrt{C_j \ell_j} + d_{j+1} \right). \tag{16}$$

Proof. The residual in Equation 15 can be written as

$$r_j(s, x_j) = \delta_j(s, x_j) + \overline{B}_{j+1}(s, x_{j+1}) - \widehat{B}_{j+1}(s, x_{j+1}).$$

Apply the triangle inequality in L^2 to Equation 15. Because ρ_j is the pushforward of ρ_k, the norm of each term evaluated on a trajectory equals its norm under ρ_j. The coverage assumption gives

$$\|\delta_j\|_{L^2(\rho_j)}^2 = \int \delta_j^2 \, \frac{d\rho_j}{d\nu_j} \, d\nu_j \le C_j \ell_j.$$

Using the analogous bound at the endpoint and adding the target-lag terms yields Equation 16.

The bound separates local regression loss ℓ_j, coverage of queried states through C_j, and lag between online and target predictions through d_j. Relating empirical losses to these population terms further requires generalization across sampled states.

A.4 Propagation of gradient error

Policy training uses the potential's input gradient. First, along differentiable transitions, the true continuation gradient obeys

$$\nabla_x B_j^\theta(s, x_j) = D_x T_j(s, x_j)^\top \nabla_x B_{j+1}^\theta(s, x_{j+1}).$$

The remaining transitions determine both the terminal evaluation point and the transport of its gradient.

Proposition 4 (Gradient error along a differentiable continuation)**.** *Under the notation of Equation 15, assume the transitions, candidate potential, and terminal critic are continuously differentiable in a neighborhood of the continuation. If $\|D_x T_j(s, x_j)\|_{\mathrm{op}} \le L_j$, $\|\nabla_x r_j(s, x_j)\| \le \xi_j$, and $\|\nabla_x e_K(s, x_K)\| \le \xi_b$, then*

$$\|\nabla_x e_k(s, x_k)\| \le \xi_b \prod_{i=k}^{K-1} L_i + \sum_{j=k}^{K-1} \xi_j \prod_{i=k}^{j-1} L_i, \tag{17}$$

where an empty product equals one.

Proof. Differentiate the identity $e_j = r_j + e_{j+1} \circ T_j$ with respect to its current input. The chain rule and the operator-norm bound give

$$\|\nabla_x e_j(s, x_j)\| \le \xi_j + L_j \|\nabla_x e_{j+1}(s, x_{j+1})\|.$$

Recursive substitution gives the local residual-gradient terms in Equation 17 and the terminal gradient error multiplied by the remaining Jacobian bounds.

For an unprojected Euler step of length h_j, a bound $\|D_x v_\theta(t_j, s, x)\|_{\mathrm{op}} \le L_v$ implies $L_j \le 1 + h_j L_v$. The product is then at most $\exp(L_v(1 - t_k))$. Projection onto the action box is nonexpansive and has operator norm at most one wherever its derivative exists. The proposition applies on regions with a fixed clipping pattern; projection is not differentiable at a clipping boundary.

Small scalar residuals need not control the gradient terms in Equation 17. In one dimension, $\varepsilon \sin(x/\varepsilon)$ has magnitude at most ε, while its derivative $\cos(x/\varepsilon)$ remains order one. Accurate continuation values alone therefore do not establish accurate value-gradient directions.

A.5 First-order relation between S and T

To prove Proposition 2, hold the sample, target velocity u, weight w, and normalization c_B fixed. Write $b(t, x) = B_{\bar{\psi}}(t, s, x)$, suppressing the state argument. When projection and gradient clipping are inactive, the guidance is $g = \nabla_x b(t, x)$.

Proof. The S and T losses, viewed as functions of the predicted velocity, are

$$\ell_S(v) = \frac{w}{d_a}\|v - u\|^2 - \lambda c_B b(t + h_S, x + h_S v),$$
$$\ell_T(v) = \frac{w}{d_a}\|v - u - \beta g\|^2.$$

Because g is detached in T, differentiation gives

$$\nabla_v \ell_S = \frac{2w}{d_a}(v - u) - \lambda c_B h_S \nabla_x b(t + h_S, x + h_S v),$$
$$\nabla_v \ell_T = \frac{2w}{d_a}(v - u - \beta g).$$

On a compact neighborhood of the sample, continuous second derivatives provide a constant M such that, for bounded v and sufficiently small h_S,

$$\|\nabla_x b(t + h_S, x + h_S v) - g\| \le M h_S (1 + \|v\|).$$

Subtract the two loss gradients, then add and subtract $\lambda c_B h_S g$. This yields

$$\nabla_v \ell_S - \nabla_v \ell_T = \left(\frac{2w\beta}{d_a} - \lambda c_B h_S\right) g + R, \qquad \|R\| \le \lambda c_B M h_S^2 (1 + \|v\|),$$

which proves Equation 10.

The objectives reveal how both updates favor the local value gradient. Expanding the successor score around the current point gives

$$-\lambda c_B b(t + h_S, x + h_S v) = C_1 - \lambda c_B h_S g^\top v + O(h_S^2), \tag{18}$$

where C_1 does not depend on v. The shifted regression term satisfies the exact identity

$$\frac{w}{d_a}\|v - u - \beta g\|_2^2 = \frac{w}{d_a}\|v - u\|_2^2 - \frac{2w\beta}{d_a} g^\top v + C_2, \tag{19}$$

with C_2 independent of v. Both updates add a linear preference in the value-gradient direction, with coefficients $\lambda c_B h_S$ and $2w\beta/d_a$. For $w > 0$, they match when $\beta = \lambda c_B h_S d_a/(2w)$. Since w varies across samples, one shared β does not generally match every sample.

For shared network parameters, the chain rule multiplies each velocity derivative by $D_\theta v_\theta^\top$ before averaging over samples. Proposition 2 concerns these per-sample loss gradients; parameter updates also depend on the velocity-network Jacobian and optimizer.

A.6 Local gain from an approximate gradient

For a fixed continuation objective $F(x) = B^\theta(t, s, x)$, the following bound characterizes the effect of a local perturbation.

Proposition 5 (Local gain from an approximate continuation gradient)**.** *Suppose F has an L-Lipschitz gradient on the segment from x to $x + \alpha g$, where $L > 0$ and $\alpha \geq 0$. If $\|g - \nabla F(x)\| \leq \varepsilon$, then*

$$F(x + \alpha g) - F(x) \geq \alpha \|g\|(\|g\| - \varepsilon) - \frac{L\alpha^2}{2}\|g\|^2. \tag{20}$$

In particular, if $\|g\| > \varepsilon$, the lower bound is positive for $0 < \alpha < 2(1 - \varepsilon/\|g\|)/L$.

Proof. Let $q = \nabla F(x)$. Integrating the gradient along the segment gives

$$F(x + \alpha g) - F(x) = \int_0^\alpha \nabla F(x + ug)^\top g \, du \geq \alpha q^\top g - \frac{L\alpha^2}{2}\|g\|^2.$$

The gradient-error bound implies $q^\top g = \|g\|^2 + (q - g)^\top g \geq \|g\|^2 - \varepsilon\|g\|$. Substituting this inequality proves Equation 20. Solving for the positive range of the resulting quadratic establishes the final statement.

For $g \neq 0$ and $r \geq 0$, let $\widehat{g} = g/\|g\|$. If F has an L-Lipschitz gradient on the segment from x to $x + r\widehat{g}$, the same argument gives

$$F(x + r\widehat{g}) - F(x) \geq r(\|g\| - \varepsilon) - \frac{Lr^2}{2}.$$

This bound separates direction quality from perturbation magnitude. For clipped guidance, g denotes the clipped vector and ε bounds its error relative to $\nabla F(x)$. The bound concerns the frozen critic-composed continuation; the experiments evaluate control returns and shared-parameter updates.

B Analysis of Feedback Design and Numerical Choices

The preceding results connect continuation prediction to local policy feedback. This section examines how weighting, projection, time sampling, and recorded traces affect that connection.

B.1 Sample reweighting and target shaping

For a fixed noise–action pair with $w > 0$, minimizing $w\|v - u\|^2/d_a$ over a free velocity gives $v = u$. Reweighting changes how this pair competes with other samples in a shared model, while target shaping changes its individual regression optimum to $u + \beta g$. To see the aggregate effect, write $X = (t, s, x_t)$ and condition on the input to the velocity field. For unrestricted square-integrable predictors and a detached weight satisfying $\mathbb{E}[w \mid X] > 0$, the conditional optimum is

$$v_D^*(X) = \frac{\mathbb{E}[w(u + \beta g(X)) \mid X]}{\mathbb{E}[w \mid X]} = \frac{\mathbb{E}[wu \mid X]}{\mathbb{E}[w \mid X]} + \beta g(X). \tag{21}$$

Setting the conditional quadratic's derivative to zero gives this optimum. For a fixed potential and state-based guidance, the correction is added to the weighted FM optimum. Reweighting and local feedback can therefore be varied as distinct factors.

State-dependent feedback also differs from independent zero-mean target noise. If ζ is independent of the sample and has zero mean and finite variance, then

$$\mathbb{E}_\zeta\|v - u - \zeta\|^2 = \|v - u\|^2 + \mathbb{E}_\zeta\|\zeta\|^2.$$

The added term is constant in v, leaving the expected regression gradient unchanged. By contrast, $g(X)$ depends on the intermediate state and enters the optimum in Equation 21. This objective-level comparison parallels the analysis of noise-injection choices in FINO (Shin et al., 2026).

B.2 Successor evaluation, projection, and combined feedback

With action projection active, define $p(v) = \mathrm{Proj}_{\mathcal{A}}(x + h_S v)$. Away from clipping boundaries, the exact S derivative is

$$\nabla_v \ell_S = \frac{2w}{d_a}(v - u) - \lambda c_B h_S D \,\mathrm{Proj}_{\mathcal{A}}(x + h_S v)^\top \nabla_x b(t_S^+, p(v)).$$

The projection Jacobian suppresses the value derivative in saturated coordinates. T uses a clipped gradient at the unprojected current interpolation and places it in a detached regression target. The interfaces therefore differ in evaluation point, weighting, and projection even when their first-order strengths are matched.

C combines the two interfaces. In the smooth small-step regime of Proposition 2, its leading guidance coefficient is $2w\beta/d_a + \lambda c_B h_S$. A higher score for C may therefore reflect greater guidance magnitude, the use of two evaluation points, or both.

B.3 Terminal connectivity in continuous-time sampling

Proposition 1 requires each transition chain to reach the supervised terminal time. The implementation samples t continuously from $[0, 1 - h]$ and advances by a fixed $h = 1/K$, while supervising the boundary at 1. A chain beginning at a generic t can end inside $(1 - h, 1)$ without reaching that boundary.

The resulting ambiguity is visible in a simple example. Let the velocity field and terminal critic be zero on an unprojected region, and choose a smooth nonzero f. Then

$$\widehat{B}(t, s, x) = \sin(2\pi K t) f(x) \tag{22}$$

has zero terminal error and satisfies $\widehat{B}(t + h, s, x) = \widehat{B}(t, s, x)$ for every sampled interior pair. Nevertheless, its values and spatial gradients can be nonzero between the terminal-connected grid points. Continuous fixed-step consistency and the terminal loss alone therefore leave an ambiguity that the fixed-grid proposition excludes.

A terminal-connected reference can query grid times or use $\Delta(t) = \min(h, 1 - t)$, shortening both the temporal and spatial final step. The reported implementation retains continuous fixed-step sampling, so the fixed-grid result does not by itself establish between-time consistency.

B.4 Difference between backup and generation samplers

Potential backups project every local successor, whereas the deployed flow projects only its endpoint. For fixed samplers, we can bound the resulting transition mismatch. Let a reference trajectory use T_j and a second trajectory use $\widetilde{T}_j$, with the same starting point at time k. Suppose T_j is L_j-Lipschitz and $\|\widetilde{T}_j(y) - T_j(y)\| \le a_j$ on the second trajectory. Writing $d_j = \|\widetilde{x}_j - x_j\|$ gives $d_{j+1} \le a_j + L_j d_j$, with $d_k = 0$, and hence

$$\|\widetilde{x}_K - x_K\| \le \sum_{j=k}^{K-1} a_j \prod_{i=j+1}^{K-1} L_i. \tag{23}$$

If the shared endpoint critic is L_Q-Lipschitz on the compared endpoints, the difference between their continuation values is bounded by L_Q times the right-hand side. This follows directly by applying the critic's Lipschitz inequality after the transition recursion.

Equation 23 isolates sampler mismatch from potential approximation error. When the samplers differ only by intermediate projection, a_j measures the displacement removed at the relevant step. Even an exact fit to one sampler's continuation can disagree with the other's.

B.5 Recorded traces and online supervision

An executed flow trace supplies several intermediate snapshots associated with one realized action. Conditional on this action, the expected environmental target is the one-step Bellman backup of the next-state actor and critic,

$$\mathbb{E}[y \mid s, a] = \mathbb{E}_{r, s', d \mid s, a}\left[r + \gamma(1 - d)\mathbb{E}_{z'} Q_{\bar{\phi}}(s', \mathrm{Proj}_{\mathcal{A}}\, f_\omega(s', z'))\right].$$

Regressing recorded intermediate points on this target links potential predictions to interaction data. The target belongs to the recorded continuation: an older flow may produce a different action from the current flow, and restarting from a clipped snapshot may change the continuation. We therefore treat trace supervision as auxiliary regression on recorded behavior. Figure 5 tests its contribution to online performance.

C Implementation Details

C.1 Training and numerical sampling

FTBP adds a time-conditioned continuation potential to GFP's critic, velocity field, and one-step actor. Each network has four 512-wide GELU layers. The critic and potential use layer normalization and two independent heads: the critic takes their minimum, while the potential takes their mean. The potential and velocity field receive a 64-dimensional sinusoidal time encoding.

All losses use the same pre-update parameter snapshot. With κ_B denoting the potential-loss weight, one optimizer step minimizes

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_Q + \mathcal{L}_{\text{actor}} + \mathcal{L}_{\text{flow}} + \kappa_B \mathcal{L}_B. \tag{24}$$

Each loss updates its designated network. In S, the target potential is fixed while its input derivative passes through the proposed successor to the velocity field. In T, the potential gradient is detached before shaping the regression target. After the optimizer step, the critic and potential targets use pre-update online parameters:

$$\bar{\phi}_{n+1} = (1-\tau_Q)\bar{\phi}_n + \tau_Q \phi_n, \qquad \bar{\psi}_{n+1} = (1-\tau_B)\bar{\psi}_n + \tau_B \psi_n.$$

The implementation projects each local backup successor but clips only the final generated action. Because interior times are sampled continuously with $h = 1/K$, the numerical learner lacks the terminal-connected grid assumed in Proposition 1. S retains its spatial step h_S when the successor time is clamped to 1; T uses the shifted velocity only as a regression target. Appendix B analyzes these choices.

C.2 Weights, potential samples, and feedback

GFP compares each replay action a with a sampled reference action a_{ref}. With detached $c_Q = (\text{mean}_{\mathcal{B}} |Q_\phi(s, \text{Proj}_{\mathcal{A}} f_\omega(s,z))|)^{-1}$, its weight is

$$w(s,a) = \frac{\exp(c_Q Q_{\bar{\phi}}(s,a)/\eta)}{\exp(c_Q Q_{\bar{\phi}}(s,a)/\eta) + \exp(c_Q Q_{\bar{\phi}}(s,a_{\text{ref}})/\eta)}.$$

The reference may include actor and flow actions; a separate $\eta = 0$ branch selects unweighted regression. Boundary supervision averages equally over replay, actor, and flow actions. Interior points come from clipped interpolation states or from a partial flow rollout of M Euler steps to time t; a Bernoulli draw selects the flow source with probability ρ. Both the sampled point and its projected local successor are detached during potential regression. Each potential head fits the same target, and their mean supplies policy feedback.

For the configuration names `local_value`, `velocity_target`, and `both`, the coefficients (β, λ) in Equation 9 are $(0, b_0 w_0)$, $(b_0, 0)$, and $(b_0, b_0 w_0)$, respectively, where b_0 and w_0 are the configured guidance and successor-weight parameters. The default operator clips a raw gradient g to $\mathcal{C}(g) = g \min\{1, c_g/(\|g\|_2 + 10^{-6})\}$ with $c_g = 1$. For S, $c_B = (\text{mean}_{\mathcal{B}} |B_{\bar{\psi}}(t_S^+, s, x_S^+)| + 10^{-6})^{-1}$ is detached.

C.3 Online traces and reference settings

A flow-generated action stores clipped snapshots, while integration continues from the unprojected internal states. Besides Equation 11, online potential training uses endpoint regression and adjacent-

trace consistency:

$$\mathcal{L}_{\text{end}} = \mathbb{E}[(B_\psi(1, s, a) - \text{sg}[y])^2],$$
$$\mathcal{L}_{\text{cons}} = \mathbb{E}\left[\frac{1}{m-1}\sum_{j=1}^{m-1}(B_\psi(t_j, s, \tilde{x}_j) - \text{sg}[B_{\bar{\psi}}(t_{j+1}, s, \tilde{x}_{j+1})])^2\right]. \tag{25}$$

For $m = 1$, the consistency term is omitted. These losses update only the potential. In the full model, trace and endpoint losses have unit inner weights; their outer trace weight is separate from κ_B:

$$\mathcal{L}_{\text{joint}}^{\text{online}} = \mathcal{L}_{\text{joint}} + \lambda_{\text{tr}}(\mathcal{L}_{\text{trace}} + \mathcal{L}_{\text{end}} + \lambda_c \mathcal{L}_{\text{cons}}). \tag{26}$$

At inference, the one-step actor needs one forward pass; the full flow needs K velocity evaluations. Neither queries the potential.

Table 3 lists the configuration shared by all FTBP variants and task families; the variants differ only in how the potential enters the flow objective, and Algorithm 2 summarizes the resulting online update.

Algorithm 1 Offline training of FTBP-S, FTBP-T, and FTBP-C.

Input: Offline dataset $\mathcal{D}$, variant $m \in \{\text{S}, \text{T}, \text{C}\}$, and networks $Q_\phi, v_\theta, f_\omega, B_\psi$.
1: Initialize target critic $Q_{\bar{\phi}}$ and target potential $B_{\bar{\psi}}$.
2: Set (β, λ) to $(0, b_0 w_0)$ for S, $(b_0, 0)$ for T, or $(b_0, b_0 w_0)$ for C.
3: **for** each offline update **do**
4: Sample $(s, a, r, s', d) \sim \mathcal{D}$, noise z, and time t; form $x_t = (1-t)z + ta$.
5: Compute critic and one-step actor losses (Equations 1 and 4).
6: Sample boundary and interior states; fit B_ψ to critic and local targets (Equation 7).
7: Form $\mathcal{L}_{\text{flow}}(\theta; \beta, \lambda)$: T shapes the target, S scores a successor, and C does both (Equation 9).
8: Update $\phi, \omega, \psi, \theta$ from their losses in Equation 24; then update $\bar{\phi}, \bar{\psi}$.
9: **end for**

Algorithm 2 Online fine-tuning of an offline-trained FTBP variant.

Input: Model from Algorithm 1, environment, online replay, and trace buffer $\mathcal{T}$.
1: **for** each online interaction **do**
2: On trace-collection steps, generate (a, τ) with the flow; otherwise generate a with the one-step actor.
3: Execute a; store the transition in online replay and, if recorded, (s, τ, a, r, s', d) in $\mathcal{T}$.
4: Sample replay and form the base objective $\mathcal{L}_{\text{joint}}$ with the chosen (β, λ) (Equation 24).
5: **if** a trace update is due and $\mathcal{T}$ can be sampled **then**
6: Compute the TD target and trace, endpoint, and consistency losses from $\mathcal{T}$.
7: Update using $\mathcal{L}_{\text{joint}}^{\text{online}}$ (Equation 26).
8: **else**
9: Update using $\mathcal{L}_{\text{joint}}$.
10: **end if**
11: Update target networks; advance to s' or reset after episode end.
12: **end for**

Table 3: Hyperparameter configuration shared by all FTBP variants and tasks.

Setting	Value
Optimizer; learning rate; batch size	Adam; 3×10^{-4}; 1024
Discount; target rates (τ_Q, τ_B)	0.995; $(0.005, 0.005)$
Flow steps K; partial steps M	10; 4
Actor coefficient α; weight temperature η	0.1; 0.1
Value reference; critic bootstrap action	Minimum of actor/flow; current actor
Guidance β; potential weight κ_B	0.03; 0.3
Boundary / local-TD weights; flow fraction ρ	$1/1$; 0.5
Local steps h, h_S; guidance cap c_g	$0.1, 0.1$; 1
Trace snapshots m; flow / trace update interval	4; every 4 interactions
Outer trace / consistency weights $(\lambda_{\mathrm{tr}}, \lambda_c)$	$(0.3, 0.25)$
Replay / trace capacities; balanced sampling	2×10^6 / 5×10^5; disabled
Offline updates; online interactions	10^6; 5×10^5

D Experimental Protocol

D.1 Toy motivation experiment

The toy dataset contains 20,000 two-dimensional actions in $[-1, 1]^2$, drawn from three equally likely Gaussian modes. The mode centers are $(-0.55, -0.35)$, $(0.5, -0.3)$, and $(0, 0.55)$, each with coordinate standard deviation 0.17. The analytic reward is $R(a) = \sum_{j=1}^{3} h_j \exp(-\|a - c_j\|_2^2/(2 \cdot 0.23^2))$, with heights $(0.35, 0.65, 1)$.

Weighted FM and FTBP-T share $w(a) = \mathrm{sigmoid}((R(a) - \bar{R}_{\mathcal{D}})/0.1)$, initialization, samples, and noise draws. Each of five paired seeds receives 20,000 updates; evaluation uses 10,000 shared noise samples per seed. The final density difference uses a common 160×160 grid, Gaussian bandwidth 0.05, and reflected boundaries. Mean final rewards are 0.605 ± 0.005 and 0.674 ± 0.003, respectively (mean $\pm$ seed standard deviation).

D.2 Benchmarks and scoring

The primary benchmark comprises five single-task goals in each of seven OGBench environments: antmaze-large, antmaze-giant, humanoidmaze-medium, humanoidmaze-large, antsoccer-arena, cube-double, and puzzle-4x4. The first five use `navigate` goals; the last two use `play`. Full identifiers append `-singletask-task<i>-v0`, with $i = 1, \ldots, 5$. Additional results cover antmaze-large-stitch, antsoccer-arena-stitch, cube-double-noisy, and puzzle-4x4-noisy, each with five goals, along with six D4RL AntMaze datasets.

Our runs use 10^6 offline updates and 5×10^5 online interactions, with evaluations every 50,000 steps over 50 episodes. The local main-table entries report the executed one-step actor at the offline endpoint and final online checkpoint. Published Q-Flow and GFP offline results instead average their final three evaluations. OGBench scores are success percentages; D4RL scores are normalized returns. All experiments are conducted with more than 10 random seeds.

E Additional Experiments

Tables 4–7 give offline and online scores for all 55 OGBench goals and six D4RL AntMaze datasets. Online scores follow fine-tuning from the corresponding offline checkpoints. Entries show mean $\pm$ seed standard deviation. The relatively large standard deviations on some tasks arise from bimodal outcomes across seeds, where training may either converge to near-perfect performance or fail to acquire a successful policy, resulting in returns close to zero.

Figures 7–9 track the executed one-step actor. Offline curves use a zero display anchor and evaluations every 0.2M updates through 1M; online curves use 0.1M-interaction intervals through 0.5M.

Table 4: **Per-task OGBench results: offline.** Bold marks the highest mean in each row.

Environment	GFP	Q-Flow	FTBP-S	FTBP-T	FTBP-C
antmaze-large-navigate-task1	95 ± 2	96 ± 4	95 ± 2	97 ± 2	**98 ± 1**
antmaze-large-navigate-task2	91 ± 4	87 ± 7	**93 ± 5**	90 ± 3	**93 ± 5**
antmaze-large-navigate-task3	**98 ± 2**	96 ± 3	96 ± 2	**98 ± 2**	**98 ± 2**
antmaze-large-navigate-task4	90 ± 3	82 ± 12	88 ± 5	92 ± 7	**93 ± 3**
antmaze-large-navigate-task5	94 ± 2	92 ± 5	94 ± 2	**96 ± 2**	96 ± 3
antmaze-large-stitch-task1	93 ± 2	**96 ± 3**	86 ± 6	93 ± 4	89 ± 5
antmaze-large-stitch-task2	81 ± 6	45 ± 28	75 ± 11	82 ± 10	**87 ± 4**
antmaze-large-stitch-task3	94 ± 3	93 ± 5	**98 ± 2**	97 ± 1	97 ± 3
antmaze-large-stitch-task4	9 ± 23	**20 ± 35**	0 ± 0	0 ± 0	0 ± 0
antmaze-large-stitch-task5	74 ± 10	**89 ± 1**	84 ± 5	82 ± 8	87 ± 5
antmaze-giant-navigate-task1	24 ± 23	9 ± 10	**34 ± 20**	13 ± 15	30 ± 16
antmaze-giant-navigate-task2	64 ± 37	38 ± 8	**84 ± 12**	50 ± 18	78 ± 7
antmaze-giant-navigate-task3	16 ± 20	16 ± 11	37 ± 18	28 ± 19	**43 ± 14**
antmaze-giant-navigate-task4	8 ± 18	**36 ± 36**	0 ± 0	1 ± 1	0 ± 0
antmaze-giant-navigate-task5	47 ± 41	74 ± 17	79 ± 12	75 ± 18	**88 ± 5**
humanoidmaze-medium-navigate-task1	**88 ± 6**	85 ± 4	86 ± 8	88 ± 9	87 ± 3
humanoidmaze-medium-navigate-task2	**96 ± 2**	94 ± 5	93 ± 3	94 ± 6	95 ± 4
humanoidmaze-medium-navigate-task3	86 ± 11	**93 ± 6**	85 ± 12	89 ± 6	88 ± 6
humanoidmaze-medium-navigate-task4	0 ± 0	**48 ± 21**	0 ± 0	0 ± 0	0 ± 0
humanoidmaze-medium-navigate-task5	97 ± 2	97 ± 3	**97 ± 1**	96 ± 4	97 ± 2
humanoidmaze-large-navigate-task1	24 ± 23	8 ± 3	**53 ± 27**	29 ± 31	49 ± 36
humanoidmaze-large-navigate-task2	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
humanoidmaze-large-navigate-task3	19 ± 10	17 ± 6	11 ± 5	18 ± 13	**20 ± 22**
humanoidmaze-large-navigate-task4	0 ± 1	**7 ± 7**	2 ± 4	0 ± 0	6 ± 8
humanoidmaze-large-navigate-task5	22 ± 28	4 ± 7	20 ± 19	6 ± 7	**26 ± 29**
antsoccer-arena-navigate-task1	82 ± 9	75 ± 7	**89 ± 9**	82 ± 5	85 ± 4
antsoccer-arena-navigate-task2	95 ± 5	**95 ± 3**	93 ± 4	95 ± 4	**95 ± 3**
antsoccer-arena-navigate-task3	54 ± 9	57 ± 6	**59 ± 3**	58 ± 13	59 ± 4
antsoccer-arena-navigate-task4	47 ± 9	17 ± 12	42 ± 14	46 ± 7	**47 ± 6**
antsoccer-arena-navigate-task5	33 ± 19	28 ± 18	29 ± 13	**41 ± 25**	38 ± 17
antsoccer-arena-stitch-task1	64 ± 4	49 ± 12	58 ± 10	**66 ± 7**	65 ± 5
antsoccer-arena-stitch-task2	**59 ± 4**	46 ± 6	52 ± 13	46 ± 20	51 ± 9
antsoccer-arena-stitch-task3	19 ± 6	15 ± 7	25 ± 4	23 ± 6	**26 ± 4**
antsoccer-arena-stitch-task4	**27 ± 8**	9 ± 5	22 ± 10	23 ± 9	24 ± 7
antsoccer-arena-stitch-task5	5 ± 4	0 ± 0	4 ± 4	2 ± 3	**9 ± 4**
cube-double-play-task1	77 ± 8	66 ± 8	71 ± 7	72 ± 5	**78 ± 8**
cube-double-play-task2	35 ± 10	26 ± 15	38 ± 11	53 ± 5	**68 ± 14**
cube-double-play-task3	**44 ± 10**	34 ± 14	36 ± 2	28 ± 15	38 ± 8
cube-double-play-task4	6 ± 5	4 ± 3	2 ± 1	4 ± 4	**6 ± 4**
cube-double-play-task5	48 ± 14	51 ± 10	**67 ± 2**	66 ± 10	63 ± 7
cube-double-noisy-task1	94 ± 7	80 ± 6	**94 ± 3**	93 ± 5	93 ± 6
cube-double-noisy-task2	74 ± 10	62 ± 13	72 ± 9	70 ± 15	**76 ± 7**
cube-double-noisy-task3	73 ± 10	59 ± 6	73 ± 14	73 ± 10	**74 ± 8**
cube-double-noisy-task4	42 ± 9	30 ± 4	39 ± 17	**43 ± 7**	35 ± 9
cube-double-noisy-task5	33 ± 12	19 ± 3	26 ± 15	**45 ± 11**	39 ± 11
puzzle-4×4-play-task1	44 ± 13	52 ± 5	56 ± 12	45 ± 9	**59 ± 11**
puzzle-4×4-play-task2	10 ± 4	**17 ± 5**	11 ± 5	12 ± 5	13 ± 5
puzzle-4×4-play-task3	44 ± 13	44 ± 10	45 ± 5	**53 ± 10**	46 ± 3
puzzle-4×4-play-task4	18 ± 6	20 ± 8	16 ± 8	**21 ± 7**	19 ± 6
puzzle-4×4-play-task5	**11 ± 6**	10 ± 3	8 ± 8	7 ± 6	10 ± 1
puzzle-4×4-noisy-task1	39 ± 9	36 ± 11	**46 ± 10**	41 ± 10	45 ± 7
puzzle-4×4-noisy-task2	1 ± 1	**8 ± 5**	2 ± 1	1 ± 1	0 ± 0
puzzle-4×4-noisy-task3	**52 ± 8**	20 ± 5	46 ± 9	44 ± 16	51 ± 11
puzzle-4×4-noisy-task4	5 ± 2	**9 ± 4**	3 ± 3	6 ± 5	3 ± 4
puzzle-4×4-noisy-task5	0 ± 0	**3 ± 2**	0 ± 1	1 ± 1	1 ± 1

Table 5: **Per-task D4RL AntMaze results: offline.** Bold marks the highest mean in each row.

Environment	GFP	FTBP-S	FTBP-T	FTBP-C
antmaze-umaze-v2	98 ± 2	98 ± 1	**100 ± 1**	99 ± 2
antmaze-umaze-diverse-v2	**95 ± 3**	89 ± 7	90 ± 5	94 ± 4
antmaze-medium-play-v2	83 ± 8	**87 ± 6**	83 ± 8	83 ± 5
antmaze-medium-diverse-v2	63 ± 14	62 ± 10	**76 ± 14**	67 ± 11
antmaze-large-play-v2	85 ± 4	86 ± 9	**89 ± 4**	87 ± 5
antmaze-large-diverse-v2	85 ± 5	84 ± 5	85 ± 8	**86 ± 7**

Table 6: **Per-task OGBench results: online.** Bold marks the highest mean in each row.

Environment	FINO	Q-Flow	FTBP-S	FTBP-T	FTBP-C
antmaze-large-navigate-task1	98 ± 1	98 ± 3	**99 ± 1**	99 ± 2	**99 ± 1**
antmaze-large-navigate-task2	92 ± 1	90 ± 4	97 ± 2	**99 ± 2**	98 ± 2
antmaze-large-navigate-task3	**100 ± 0**	99 ± 1	**100 ± 0**	**100 ± 0**	**100 ± 0**
antmaze-large-navigate-task4	**99 ± 1**	94 ± 3	**99 ± 1**	**99 ± 1**	**99 ± 1**
antmaze-large-navigate-task5	**100 ± 1**	97 ± 2	98 ± 2	99 ± 1	99 ± 1
antmaze-large-stitch-task1	0 ± 0	99 ± 2	**100 ± 0**	98 ± 2	99 ± 1
antmaze-large-stitch-task2	0 ± 0	89 ± 7	96 ± 1	96 ± 3	**97 ± 3**
antmaze-large-stitch-task3	99 ± 1	100 ± 1	**100 ± 0**	**100 ± 0**	**100 ± 0**
antmaze-large-stitch-task4	3 ± 2	99 ± 2	99 ± 1	99 ± 1	**100 ± 0**
antmaze-large-stitch-task5	0 ± 0	95 ± 3	**99 ± 1**	**99 ± 1**	**99 ± 1**
antmaze-giant-navigate-task1	87 ± 7	91 ± 5	**94 ± 7**	78 ± 6	88 ± 5
antmaze-giant-navigate-task2	**99 ± 1**	94 ± 3	97 ± 2	**99 ± 1**	**99 ± 1**
antmaze-giant-navigate-task3	0 ± 0	72 ± 22	59 ± 8	57 ± 12	**81 ± 5**
antmaze-giant-navigate-task4	80 ± 6	72 ± 40	91 ± 3	90 ± 7	**93 ± 5**
antmaze-giant-navigate-task5	99 ± 1	96 ± 4	99 ± 1	**100 ± 0**	**100 ± 0**
humanoidmaze-medium-navigate-task1	93 ± 4	**97 ± 5**	92 ± 5	94 ± 4	96 ± 3
humanoidmaze-medium-navigate-task2	**100 ± 0**	98 ± 2	97 ± 2	98 ± 3	99 ± 1
humanoidmaze-medium-navigate-task3	74 ± 23	98 ± 2	98 ± 1	99 ± 1	**100 ± 1**
humanoidmaze-medium-navigate-task4	65 ± 27	**72 ± 19**	3 ± 4	0 ± 0	0 ± 0
humanoidmaze-medium-navigate-task5	98 ± 1	99 ± 1	99 ± 1	**100 ± 1**	99 ± 1
humanoidmaze-large-navigate-task1	0 ± 0	15 ± 7	49 ± 52	**50 ± 53**	49 ± 52
humanoidmaze-large-navigate-task2	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
humanoidmaze-large-navigate-task3	16 ± 48	22 ± 3	25 ± 46	25 ± 45	**33 ± 47**
humanoidmaze-large-navigate-task4	1 ± 0	9 ± 5	24 ± 45	24 ± 45	**36 ± 49**
humanoidmaze-large-navigate-task5	0 ± 0	6 ± 2	35 ± 49	26 ± 44	**43 ± 45**
antsoccer-arena-navigate-task1	83 ± 8	94 ± 3	**98 ± 1**	98 ± 2	98 ± 2
antsoccer-arena-navigate-task2	98 ± 2	97 ± 2	**100 ± 1**	99 ± 1	99 ± 1
antsoccer-arena-navigate-task3	76 ± 11	87 ± 8	92 ± 7	86 ± 7	**93 ± 4**
antsoccer-arena-navigate-task4	60 ± 8	64 ± 8	79 ± 4	**87 ± 6**	81 ± 5
antsoccer-arena-navigate-task5	35 ± 16	70 ± 8	**86 ± 5**	85 ± 9	85 ± 5
antsoccer-arena-stitch-task1	44 ± 9	73 ± 13	91 ± 3	**92 ± 6**	89 ± 7
antsoccer-arena-stitch-task2	7 ± 8	89 ± 6	91 ± 5	**93 ± 5**	90 ± 4
antsoccer-arena-stitch-task3	11 ± 6	42 ± 9	**55 ± 7**	54 ± 10	52 ± 5
antsoccer-arena-stitch-task4	11 ± 3	37 ± 17	53 ± 7	53 ± 12	**54 ± 7**
antsoccer-arena-stitch-task5	3 ± 2	32 ± 6	50 ± 8	52 ± 4	**57 ± 11**
cube-double-play-task1	98 ± 7	**99 ± 2**	94 ± 7	96 ± 3	98 ± 2
cube-double-play-task2	88 ± 5	76 ± 12	**99 ± 2**	91 ± 1	94 ± 5
cube-double-play-task3	95 ± 6	85 ± 5	93 ± 9	**97 ± 4**	94 ± 8
cube-double-play-task4	4 ± 5	15 ± 9	**18 ± 7**	6 ± 6	9 ± 6
cube-double-play-task5	93 ± 3	93 ± 4	91 ± 1	**98 ± 1**	97 ± 2
cube-double-noisy-task1	**100 ± 0**	99 ± 1	**100 ± 0**	**100 ± 0**	**100 ± 0**
cube-double-noisy-task2	27 ± 12	99 ± 1	**100 ± 0**	**100 ± 0**	**100 ± 0**
cube-double-noisy-task3	45 ± 14	**100 ± 0**	**100 ± 0**	**100 ± 0**	100 ± 1
cube-double-noisy-task4	12 ± 7	97 ± 4	**100 ± 0**	**100 ± 0**	**100 ± 0**
cube-double-noisy-task5	34 ± 10	99 ± 1	**100 ± 0**	**100 ± 0**	**100 ± 0**
puzzle-4×4-play-task1	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0
puzzle-4×4-play-task2	0 ± 0	0 ± 0	1 ± 2	50 ± 55	**58 ± 53**
puzzle-4×4-play-task3	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0
puzzle-4×4-play-task4	**80 ± 45**	0 ± 0	60 ± 55	**80 ± 45**	**80 ± 45**
puzzle-4×4-play-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4×4-noisy-task1	47 ± 8	**100 ± 0**	99 ± 1	99 ± 1	**100 ± 0**
puzzle-4×4-noisy-task2	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4×4-noisy-task3	4 ± 3	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**
puzzle-4×4-noisy-task4	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4×4-noisy-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0

Table 7: **Per-task D4RL AntMaze results: online.** Bold marks the highest mean in each row.

Environment	FINO	FTBP-S	FTBP-T	FTBP-C
antmaze-umaze-v2	99 ± 1	99 ± 1	**100 ± 1**	99 ± 2
antmaze-umaze-diverse-v2	98 ± 2	99 ± 2	**99 ± 1**	**99 ± 1**
antmaze-medium-play-v2	95 ± 2	**99 ± 1**	98 ± 2	**99 ± 1**
antmaze-medium-diverse-v2	96 ± 3	99 ± 2	98 ± 3	**99 ± 1**
antmaze-large-play-v2	94 ± 2	93 ± 2	**97 ± 2**	94 ± 2
antmaze-large-diverse-v2	91 ± 3	92 ± 1	**96 ± 3**	95 ± 3

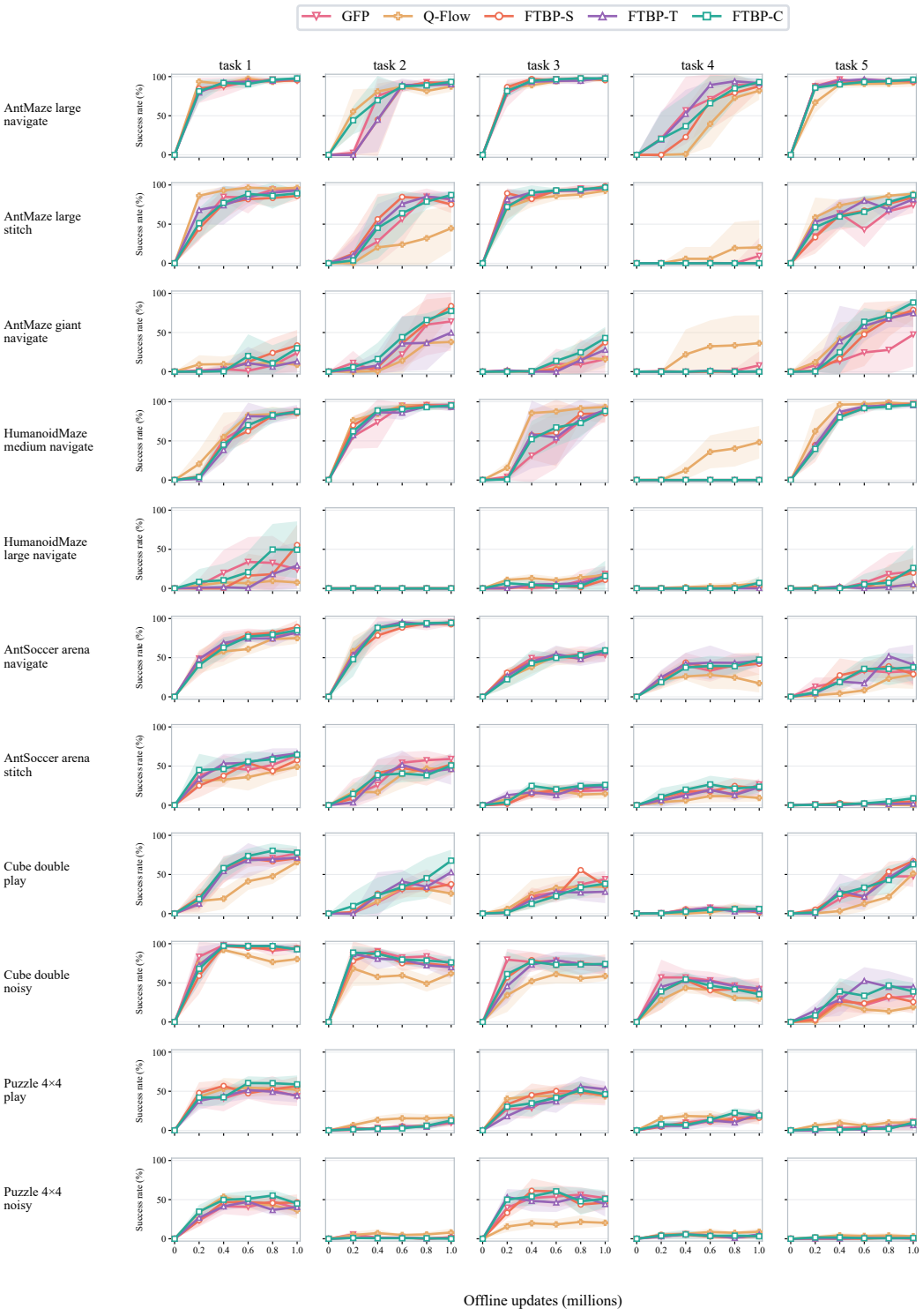


Figure 7: **Offline learning on 55 OGBench tasks.**

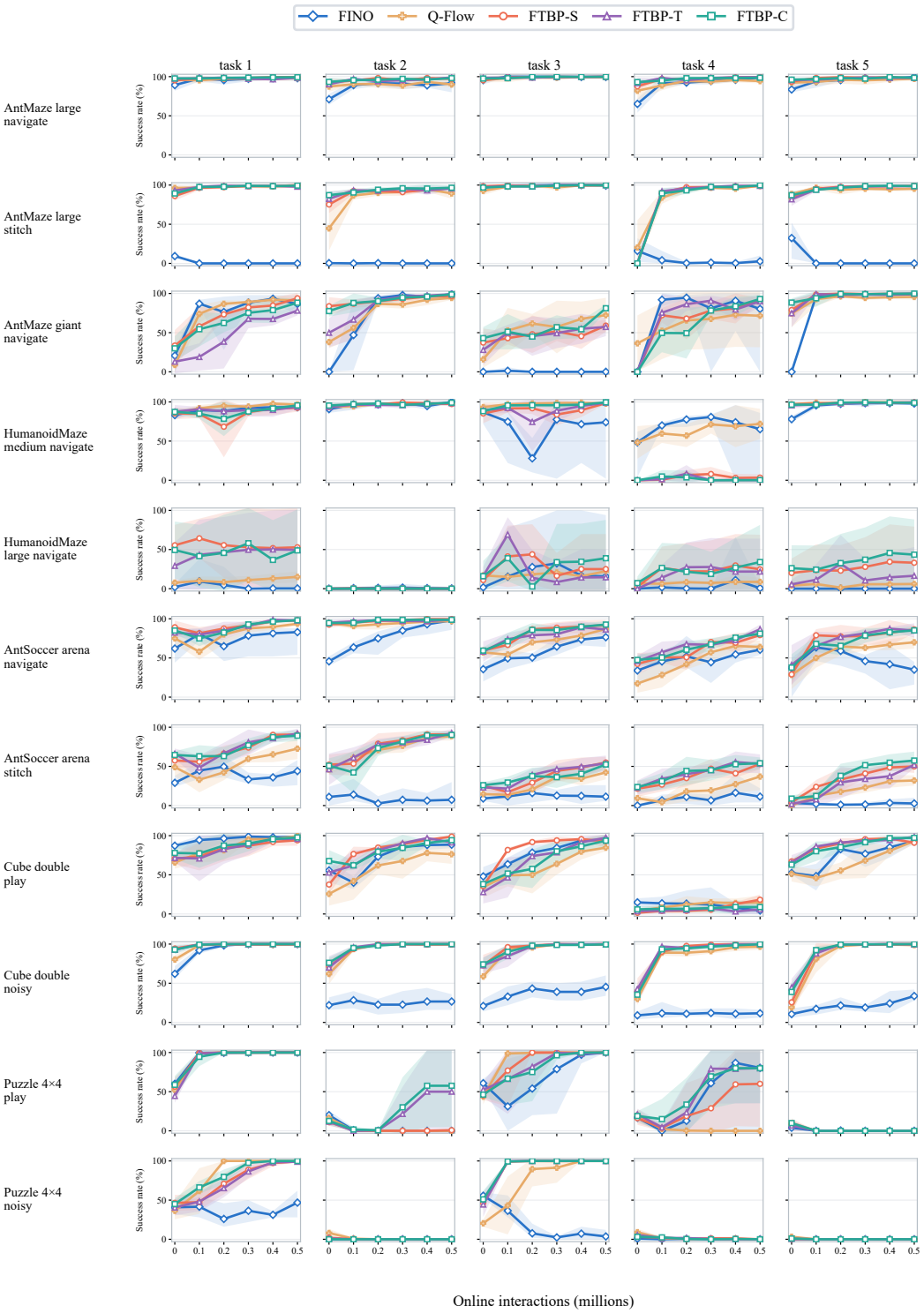


Figure 8: **Online fine-tuning on the same OGBench tasks.**

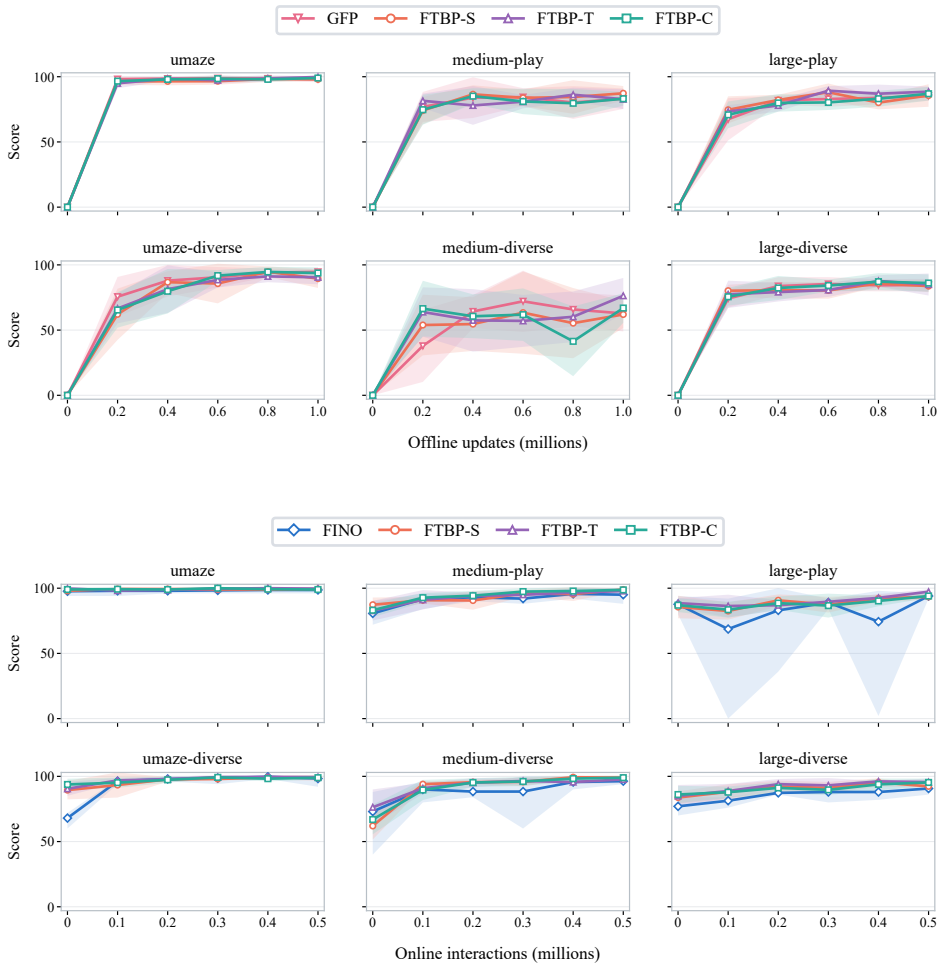


Figure 9: **D4RL AntMaze learning curves.** Offline (top grid) and online (bottom grid).

F LIMITATIONS AND SCOPE

FTBP's feedback depends on an environmental critic. Critic error can therefore enter both endpoint supervision and policy extraction. The theoretical bounds concern a fixed continuation and identify coverage, target lag, derivative error, and sampler mismatch as separate error sources. During joint training these quantities change together, so end-to-end performance is assessed empirically.

The continuous fixed-step learner admits the terminal-connectivity ambiguity illustrated in Equation 22. Clipped training states can also differ from unprojected generation states, and stored online traces can come from an earlier policy. These effects lie outside the fixed-continuation guarantee. Local bootstrapping reduces the work per interior target but adds a potential network and its losses; total computational cost depends on the training configuration.

G RELATED WORK

Value learning inside generative policies. Prior work treats generation as an internal decision process (Ren et al., 2025; Gao et al., 2025). BDPO learns a diffusion-time value through local TD, with an environmental Q boundary and pathwise KL costs, and improves the policy through successor values. FTBP instead uses deterministic flow transitions and learns terminal-value predictions separately from its value-weighted FM objective, rather than transporting a KL-regularized return. Q-Flow learns an unregularized continuation value from completed suffixes and uses its gradient to shift the FM target (Doo et al., 2026). FTBP-T shares that policy interface but learns the potential from local targets. RQL reconstructs reverse-flow trajectories, learns intermediate values with multi-step expectile targets, and optimizes successor values with FM regularization (Oberai et al., 2026). FTBP-S uses a related successor-value interface with a separately anchored critic and local targets on sampled intermediate states. These relationships motivate separate comparisons of target estimation and policy feedback.

Value-gradient guidance and alignment. Adjoint matching and QAM derive policy-improvement signals from stochastic optimal control; VGG-Flow learns a gradient field using HJB consistency and terminal gradient supervision (Domingo-Enrich et al., 2024; Li & Levine, 2026; Liu et al., 2025b). FTBP predicts the scalar value of a fixed continuation and differentiates that predictor to update the flow. General guidance theory, Q-VGM, and VINE offer complementary views of local directions and sampler-dependent value optimization (Feng et al., 2025; Wang et al., 2026; Yang et al., 2026b). TRQAM constrains path-space deviation to address critic-induced instability (Dong et al., 2026). Against these control-sensitivity and policy-change approaches, FTBP studies locally bootstrapped scalar prediction and compares the policy interfaces that use it.

Offline learning and online adaptation. FQL provides the one-step extraction backbone, and GFP provides value-aware FM weights (Park et al., 2025b; Tiofack et al., 2025). Energy-weighted FM, FlowQ, and Flow Actor-Critic combine expressive policies with value information in other ways (Zhang et al., 2025a; Alles et al., 2025; Chae et al., 2026). OFQL and bootstrapped Flow Q-Learning address action-generation cost; the latter bootstraps flow displacements instead of internal scalar values (Nguyen & Yoo, 2026; Nguyen et al., 2026). In O2O RL, Cal-QL and RLPD learn from prior data, while FINO prepares a generative policy for exploration through injected noise and entropy-guided sampling (Nakamoto et al., 2023; Ball et al., 2023; Shin et al., 2026). FTBP uses observed transitions to supervise its potential within the existing exploration scheme.

G.1 FURTHER CONNECTIONS TO GENERATIVE REINFORCEMENT LEARNING

Policy learning and extraction. Diffusion-based offline RL uses constrained actor–critic objectives, Q-score matching, and in-sample guidance (Fang et al., 2024; Psenka et al., 2024; Mao et al., 2024). Flow-based single-step completion and dispersive MeanFlow pursue efficient action generation (Koirala & Fleming, 2025; Zou et al., 2026). Other work studies flow policies for discrete actions, confounding, and safety (Khan et al., 2026; Li et al., 2026a; Tayal & Tayal, 2026). Direct differentiation through the sampler offers another route to flow-policy improvement (Yang et al., 2025). CQL, QAM, Flow Actor-Critic, and OFQL provide further reference points for conservative learning, control-based guidance, and policy extraction (Kumar et al., 2020; Li & Levine, 2026;

Chae et al., 2026; Nguyen & Yoo, 2026). These approaches frame the broader setting in which we isolate local value feedback.

Guidance during generation. Flow-map alignment, QPILOTS, test-time gradient guidance, Guided Action Flow, and optimal-transport Q-learning guide or accelerate generation through different mechanisms (Ziakas et al., 2026; Ruan et al., 2026; Zhou et al., 2026; Yang et al., 2026a; Sochopoulos et al., 2026). Stepwise Flow-GRPO assigns reward across flow steps, while potential-guided flow matching adds task-specific potential predictions (Savani et al., 2026; Mei et al., 2026). FTBP uses a scalar potential anchored to an environmental action-value critic to guide flow training.

Online optimization. Online diffusion optimization and flow-policy methods seek policy gradients or tractable training objectives through a range of designs (Ma et al., 2025; Lv et al., 2025; Liu et al., 2025a; Zhang et al., 2025b; McAllister et al., 2025; Li et al., 2026b; Zhong et al., 2026; Qiu et al., 2026; Lei et al., 2026; Shi et al., 2026). FTBP's online extension supervises the same intermediate potential used offline while retaining the existing exploration distribution and requiring no new likelihood estimator.

Flow-based environmental critics. FloQ and subsequent analysis use flow-matching parameterizations to learn environmental values (Agrawalla et al., 2025; 2026). Distributional critics and expressive value representations offer other combinations of generative policies and value learning (Chen et al., 2025; Espinosa-Dice et al., 2025; Zhong et al., 2025). FTBP keeps a conventional environmental critic and learns a separate value over internal generation states, separating the two prediction problems.