

# Posterior-Averaged Bellman Flow Matching for Distributional Critics

Zeqiao Li<sup>1</sup>, Yijing Wang<sup>1</sup>, Haoyu Wang<sup>1,2</sup>,  
Zheng Li<sup>1</sup>, Zhiqiang Zuo<sup>1</sup>

<sup>1</sup>Tianjin Key Laboratory of Intelligent Unmanned Swarm Technology and System,  
School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

<sup>2</sup>Key Laboratory of System Control and Information Processing,  
Ministry of Education of China, Shanghai 200240, China  
{lizeqiao, yjwang, why2014, zhengl, zqzuo}@tju.edu.cn

## Abstract

Distributional critics model the full return distribution, providing richer signals for exploration and risk-sensitive control. Recent work has shown that flow-based critics can model these distributions flexibly and provide effective supervision across flow time for temporal-difference learning. However, unlike standard flow matching with fixed data endpoints, a flow critic is trained on Bellman targets recursively generated by a learned target critic under the current policy. Existing objectives therefore either use noisy single-endpoint supervision or rely on the learned flow itself for additional guidance. Our key observation is that an intermediate bridge state may be compatible with several Bellman endpoints, so its velocity should average their directions under the corresponding posterior. We propose Posterior-Averaged Bellman Flow Matching (PA-BFM), which samples an endpoint pool, infers the closed-form posterior over the latent anchor, and regresses to the posterior-mean velocity. Uniform anchor selection makes this target an exact finite-pool Rao–Blackwellization of the single-endpoint target: for every  $K$ , it preserves the population target, cannot increase conditional variance, and its conditional variance scales as  $O(K^{-1})$ . The endpoint pool supports conservative updates, while the twin return flows provide critic-disagreement and unsupported-mass diagnostics. Controlled studies show that PA-BFM reduces fixed-query target mean-squared error by 79.7% and target-induced gradient variance by up to 98.0%, while improving return-distribution fidelity. Across 50 OGBench tasks and four offline-to-online scenarios, PA-BFM achieves higher aggregate performance than Value Flows.

## 1 Introduction

In actor–critic reinforcement learning (RL), policy improvement depends on the critic used to evaluate actions and construct Bellman targets. Most methods retain only the expected return, whereas distributional RL preserves information about stochastic outcomes, multimodality, and risk (Bellemare et al., 2017; Dabney et al., 2018a,b; Duan et al., 2022). Recent flow-based critics represent this distribution through a continuous transport from simple noise to return samples (Dong et al., 2026; Xu et al., 2026; Ding et al., 2026; Zhong et al., 2025; Chen et al., 2025). Flow-time supervision further provides dense signals for changing temporal-difference targets (Lipman et al., 2023; Albergo and Vanden-Eijnden, 2023; Liu et al., 2023).

Existing flow critics use two complementary forms of Bellman supervision. Endpoint matching is anchored to a sampled Bellman return but uses one continuation, whereas pathwise matching provides denser supervision through another learned successor field. Value Flows combines these objectives with target networks and confidence weighting, while Path-Coupled Bellman Flows

(PCBF) introduces source-consistent geometry and a learned-field control variate (Dong et al., 2026; Xu et al., 2026). Related bridge, distillation, and architectural corrections retain the same tension between an anchored but noisy target and guidance from a learned flow (Ding et al., 2026; Chen et al., 2025; Agrawalla et al., 2025).

We recast this tension as a conditional estimation problem. At a fixed intermediate bridge state, multiple Bellman endpoints can be compatible with the same observation. Each compatible endpoint implies a candidate velocity, and the population bridge velocity is the posterior average of these endpoint directions. The uncertainty is therefore not whether a Bellman endpoint law exists, but which compatible endpoint explains the observed intermediate state.

Our goal is to estimate this posterior velocity directly from Bellman endpoint samples. This estimator is particularly natural in distributional RL: conditioned on a replay transition, the target policy and return flow directly sample the conditional Bellman endpoint law. For the one-dimensional linear Gaussian critic bridge, candidate likelihoods are closed form, so the finite endpoint pool induces a tractable posterior over the latent anchor without endpoint retrieval or an auxiliary proposal distribution. In contrast, posterior augmentation in generative modeling often requires retrieval or approximate candidate construction, and policy-side reverse matching may require a separate proposal (Stoica et al., 2026; Yang et al., 2026; Li et al., 2026).

Based on this observation, we propose *Posterior-Averaged Bellman Flow Matching* (PA-BFM). For each transition, PA-BFM samples a finite endpoint pool, selects one endpoint uniformly to construct the bridge state, and evaluates every candidate using its Gaussian bridge likelihood. The posterior-weighted direction is exactly the conditional expectation of the single-endpoint Bellman conditional flow matching (BCFM) target. Hence every finite pool preserves the population bridge velocity and cannot increase conditional target variance. We further show that its conditional variance scales as  $O(K^{-1})$  and derive implications for gradient variability, terminal transport error, and Bellman tracking.

The endpoint posterior supports conservative reweighting, while the twin return flows provide disagreement and unsupported-mass diagnostics (Kuznetsov et al., 2020; Kumar et al., 2019, 2020; Ma et al., 2021; Fujimoto et al., 2018). The main experiments use the undistorted target and retain matched flow-critic policy extraction. Our main contributions are:

- **Posterior-averaged Bellman supervision.** We formulate source-consistent endpoint matching as posterior velocity estimation and estimate it from  $K$  Bellman endpoints using closed-form bridge posteriors, without regressing to a learned successor field.
- **Exact finite-pool guarantees.** Uniform anchor selection makes PA-BFM an exact Rao–Blackwellization of BCFM. We prove finite- $K$  unbiasedness and variance dominance, characterize  $K^{-1}$  conditional-variance scaling with an ESS-based bound, and propagate velocity error to terminal Wasserstein and Bellman tracking errors.
- **Extensions and empirical validation.** The endpoint pool supports conservative posterior tilting, while the twin return flows provide critic-disagreement and unsupported-mass diagnostics. Controlled studies, 50 OGBench tasks, and four offline-to-online scenarios show gains in estimation, return-law fidelity, and downstream performance.

## 2 Preliminaries

We review distributional returns, one-dimensional flow matching, and the source-consistent Bellman bridge used by our critic.

## 2.1 MDPs and Distributional Returns

We consider a discounted Markov decision process  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \mathcal{R}, \rho_0, \gamma)$ , where  $\mathcal{S}$  and  $\mathcal{A}$  are the state and action spaces,  $P$  is the transition kernel,  $\mathcal{R}$  is the reward function,  $\rho_0$  is the initial-state distribution, and  $\gamma \in (0, 1)$ . In offline reinforcement learning (RL), the learner observes a fixed dataset  $\mathcal{D} = \{(s, a, r, s', d)\}$ , where  $d \in \{0, 1\}$  indicates termination and  $r$  is the realized reward. We use  $h$  for environment time and  $t \in [0, 1]$  for flow time.

For a policy  $\pi$ , the return random variable is

$$Z^\pi(s, a) = \sum_{h=0}^{\infty} \gamma^h \mathcal{R}(S_h, A_h), \quad S_0 = s, \quad A_0 = a, \quad (1)$$

where  $S_{h+1} \sim P(\cdot \mid S_h, A_h)$  and  $A_h \sim \pi(\cdot \mid S_h)$  for  $h \geq 1$ . Its expectation is the action value  $Q^\pi(s, a) = \mathbb{E}[Z^\pi(s, a)]$ , whereas distributional RL models the law of  $Z^\pi(s, a)$  itself (Bellemare et al., 2017; Dabney et al., 2018a,b).

For a state-action conditional return variable  $Z$ , the distributional Bellman operator is

$$\begin{aligned} (\mathcal{T}^\pi Z)(s, a) &\stackrel{d}{=} r + \gamma(1 - d)Z(S', A'), \\ S' &\sim P(\cdot \mid s, a), \quad A' \sim \pi(\cdot \mid S'). \end{aligned} \quad (2)$$

For a nonterminal transition with reward  $r$ , its density satisfies

$$p_{\mathcal{T}^\pi Z}(z \mid s, a) = \frac{1}{\gamma} \mathbb{E}_{S', A'} \left[ p_Z \left( \frac{z - r}{\gamma} \mid S', A' \right) \right]. \quad (3)$$

Terminal transitions place all mass at  $z = r$ . The operator is a  $\gamma$ -contraction under Wasserstein metrics and has  $Z^\pi$  as its unique fixed point (Bellemare et al., 2017).

## 2.2 Flow Matching

Flow matching (FM) learns a time-dependent vector field that transports a standard Gaussian source to a target distribution  $p_1(\cdot \mid c)$ . Let  $\epsilon \sim \mathcal{N}(0, 1)$ , and let  $p_0$  denote its density. The flow solves

$$\frac{d}{dt} \phi_\theta(\epsilon \mid t, c) = v_\theta(\phi_\theta(\epsilon \mid t, c), t; c), \quad \phi_\theta(\epsilon \mid 0, c) = \epsilon. \quad (4)$$

Conditional flow matching (CFM) trains this field using tractable paths (Lipman et al., 2023; Albergo and Vanden-Eijnden, 2023; Liu et al., 2023). For independent  $\epsilon \sim \mathcal{N}(0, 1)$ ,  $y \sim p_1(\cdot \mid c)$ , and  $t \sim \text{Unif}([0, 1])$ , define

$$X_t = (1 - t)\epsilon + ty, \quad u_t(X_t, y) = y - \epsilon = \frac{y - X_t}{1 - t}. \quad (5)$$

The CFM objective is

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E} \left[ \|v_\theta(X_t, t; c) - u_t(X_t, y)\|_2^2 \right]. \quad (6)$$

Its population minimizer is  $v^*(x, t; c) = \mathbb{E}[u_t(x, Y) \mid X_t = x, c]$ : the conditional average of all path directions compatible with the observed intermediate point. For our critic,  $c = (s, a)$ , and  $Z_\theta(s, a) \stackrel{d}{=} \phi_\theta(\epsilon \mid 1, s, a)$ .

### 2.3 Bellman Endpoint Laws and Source-Consistent Bridges

Fix  $(s, a)$  and let  $\tau = (r, s', d)$  denote one transition outcome. Let  $Y \sim \mu_\tau$  denote the abstract conditional Bellman endpoint law. A sample from this law is generated by a target-flow return  $Z_{\bar{\theta}}(s', a'; \epsilon')$ :

$$\begin{aligned} Y &= r + \gamma(1 - d)Z_{\bar{\theta}}(s', A'; \epsilon'), \\ \mu_\tau &= \text{Law}(Y \mid \tau), \quad A' \sim \pi(\cdot \mid s'), \quad \epsilon' \sim \mathcal{N}(0, 1). \end{aligned} \quad (7)$$

In our implementation,  $\mu_\tau$  is the law of the shared-noise mean of the two target critics; the explicit construction is given in Appendix C.1. An endpoint pool averages the successor-action and target-flow randomness within one transition; replay sampling performs the outer average over  $\tau \mid s, a$ .

The Bellman endpoint law does not uniquely determine an intermediate path. For a nonterminal transition ( $d = 0$ ), applying the Bellman affine map to each slice of a successor flow gives  $X_t^{\text{sl}} = r + \gamma X'_t$ . If  $X'_t$  has density  $p'_t(\cdot \mid s', A')$  and velocity  $v'_t$ , then, with  $\xi = (x - r)/\gamma$ ,

$$q_t^{\text{sl}}(x) = \frac{1}{\gamma} \mathbb{E}_{A'}[p'_t(\xi \mid s', A')], \quad (8)$$

$$v_t^{\text{sl}}(x) = \gamma \frac{\mathbb{E}_{A'}[p'_t(\xi \mid s', A') v'_t(\xi \mid s', A')]}{\mathbb{E}_{A'}[p'_t(\xi \mid s', A')]} \quad (9)$$

The factor  $\gamma$  follows from  $\dot{X}_t^{\text{sl}} = \gamma \dot{X}'_t$ . Although this path reaches the correct Bellman endpoint, its source is  $\mathcal{N}(r, \gamma^2)$ , rather than the common critic source  $\mathcal{N}(0, 1)$ . For  $d = 1$ , the endpoint law is the point mass at  $Y = r$ .

We therefore use the source-consistent bridge

$$\begin{aligned} X_t &= (1 - t)\epsilon + tY, \quad \epsilon \perp Y, \\ \epsilon &\sim \mathcal{N}(0, 1), \quad Y \sim \mu_\tau. \end{aligned} \quad (10)$$

If the successor uses the linear path  $X'_t = (1 - t)\epsilon + tX'_1$  with the same source noise, then

$$X_t = X_t^{\text{sl}} + (1 - t)[(1 - \gamma)\epsilon - r]. \quad (11)$$

Thus the two paths share the endpoint at  $t = 1$ , but not their intermediate marginals.

Using the standard Gaussian density  $p_0$ , define

$$\ell_t(x \mid y) = \frac{1}{1 - t} p_0\left(\frac{x - ty}{1 - t}\right), \quad u_t(x, y) = \frac{y - x}{1 - t}. \quad (12)$$

The transition-conditional bridge velocity is

$$v_\tau^*(x, t) = \mathbb{E}[u_t(x, Y) \mid X_t = x, \tau]. \quad (13)$$

The full state-action velocity is recovered by the outer transition mixture,

$$v^*(x, t \mid s, a) = \mathbb{E}[v_\tau^*(x, t) \mid X_t = x, s, a]. \quad (14)$$

Hence the endpoint pool estimates the velocity conditional on one transition, while replay sampling accounts for environment-transition randomness.

## 3 Method

We use PA-BFM for the complete algorithm and PA-BCFM for its posterior-averaged critic-training objective.

### 3.1 Bellman Objectives and Posterior Velocity

Bellman target noise is recursively reused in later temporal-difference updates. Unlike standard FM with fixed data endpoints, a flow critic draws from a learned Bellman law under the current policy, so error can re-enter later targets. We seek an endpoint-anchored estimator that averages continuation uncertainty already present in this law. For a sampled transition  $\tau = (r, s', d)$ , draw a Bellman endpoint

$$\begin{aligned} Y &= r + \gamma(1 - d)Z_{\bar{\theta}}(s', A'; \epsilon'), \\ A' &\sim \pi(\cdot | s'), \quad \epsilon' \sim \mathcal{N}(0, 1), \end{aligned} \tag{15}$$

and construct the source-consistent bridge  $X_t = (1 - t)\epsilon + tY$ , whose supervision direction is  $u_t(X_t, Y) = (Y - X_t)/(1 - t) = Y - \epsilon$ . Bellman conditional flow matching (BCFM) minimizes

$$\mathcal{L}_{\text{BCFM}}(\theta) = \mathbb{E}[\|v_{\theta}(X_t, t; s, a) - u_t(X_t, Y)\|_2^2]. \tag{16}$$

Conditioned on  $X_t = x$ , its population target is

$$v_{\tau}^*(x, t) = \mathbb{E}[u_t(x, Y) | X_t = x, \tau]. \tag{17}$$

BCFM is conditionally unbiased, but  $X_t = x$  does not identify which compatible endpoint was sampled. Using one continuation therefore introduces avoidable label variance. This is a latent-identity problem: after observing the bridge point, the missing quantity is which compatible continuation produced it.

Value Flows additionally uses distributional conditional flow matching (DCFM) to impose pathwise consistency (Dong et al., 2026). With  $X'_t = \phi_{\bar{\theta}}(\epsilon' | t, s', A')$ , its practical objective is

$$\begin{aligned} \mathcal{L}_{\text{DCFM}}^{\text{VF}}(\theta) &= \mathbb{E}[\|\Delta_t^{\text{VF}}\|_2^2], \\ \Delta_t^{\text{VF}} &= v_{\theta}(r + \gamma X'_t, t; s, a) - v_{\bar{\theta}}(X'_t, t; s', A'). \end{aligned} \tag{18}$$

DCFM provides dense learned-field supervision, but it is not an estimator of the source-consistent posterior target in Eq. (17). The corresponding slice-wise affine geometry would instead require

$$v_{\theta}(r + \gamma x', t; s, a) \approx \gamma v_{\bar{\theta}}(x', t; s', A'), \tag{19}$$

and follows the source-mismatched path in Eq. (9). Because both sides of Eq. (18) are learned, degenerate shared fields may also satisfy its residual. DCFM is therefore dense but self-referential, whereas BCFM is anchored but single-sample. Bridge corrections and critic-specific architectures can strengthen this supervision, but preserve the distinction between sampled endpoint anchors and guidance from a learned field (Ding et al., 2026; Chen et al., 2025; Agrawalla et al., 2025). PCBF instead corrects the single-endpoint target with a learned-field control variate and a tunable coefficient (Xu et al., 2026). Thus BCFM samples one continuation, PCBF corrects it with a learned field, and PA-BFM conditions the same target on a finite endpoint pool. This comparison is estimator-level: PA-BFM changes how the Bellman direction is formed while leaving the target policy, return-flow parameterization, and policy-extraction protocol unchanged.

### 3.2 Posterior-Averaged Bellman Flow Matching

PA-BCFM implements the conditional averaging suggested above in four steps: it samples a Bellman endpoint pool, selects one latent anchor to form the bridge point, infers the anchor posterior, and averages the corresponding BCFM directions. Unlike generic posterior augmentation, no retrieval

model is needed: the target return flow already supplies candidates from the relevant transition-conditional law, and the linear Gaussian bridge supplies their likelihoods analytically. The pool exposes uncertainty about compatible continuations without adding another learned correction field.

For one sampled transition, PA-BCFM draws

$$\mathbf{Y} = (Y_1, \dots, Y_K), \quad Y_j \stackrel{\text{i.i.d.}}{\sim} \mu_\tau, \quad j = 1, \dots, K. \quad (20)$$

Each endpoint is generated by the target policy and target return flow as in Eq. (15). In practice,  $\mu_\tau$  is instantiated by the shared-noise mean of the twin target critics; this does not change the estimator below. All candidates share the sampled transition but retain independent successor-action and return-noise draws, so increasing  $K$  targets continuation uncertainty rather than replay-transition uncertainty.

It then chooses one candidate uniformly to construct the observed bridge point,

$$\begin{aligned} I &\sim \text{Unif}\{1, \dots, K\}, & \epsilon &\sim \mathcal{N}(0, 1), \\ x_t &= (1 - t)\epsilon + tY_I. \end{aligned} \quad (21)$$

The index  $I$  is the unobserved identity of the endpoint that generated  $x_t$ . Sampling the anchor from the same pool gives a finite candidate model in which the posterior responsibility of every endpoint can be evaluated exactly.

Given the bridge point, a candidate  $Y_j$  explains  $x_t$  through the source value  $(x_t - tY_j)/(1 - t)$ . Bayes' rule therefore gives

$$\begin{aligned} w_j(x_t, t) &= \frac{\ell_t(x_t | Y_j)}{\sum_{\ell=1}^K \ell_t(x_t | Y_\ell)} \\ &= \Pr(I = j | X_t = x_t, \mathbf{Y}, \tau, t). \end{aligned} \quad (22)$$

Thus  $w_j$  is the exact posterior responsibility of candidate  $Y_j$ , not a heuristic similarity score. Uniform averaging ignores this compatibility by assigning every candidate the same responsibility. Early in the bridge, many endpoints can explain  $x_t$  and the weights are broad; near the endpoint, compatibility becomes sharper. The endpoint-pool effective sample size below measures this time-dependent concentration.

Finally, define the candidate directions and posterior target by

$$\begin{aligned} u_j &= u_t(x_t, Y_j) = \frac{Y_j - x_t}{1 - t}, \\ \hat{u}_K(x_t, t) &= \sum_{j=1}^K w_j(x_t, t) u_j. \end{aligned} \quad (23)$$

PA-BCFM trains the twin online critics toward this shared target:

$$\begin{aligned} \mathcal{L}_{\text{PA-BCFM}} &= \frac{1}{2} \sum_{c=1}^2 \mathbb{E} \left[ \left\| v_{\theta_c}(x_t, t; s, a) - \text{sg}[\hat{u}_K(x_t, t)] \right\|_2^2 \right], \end{aligned} \quad (24)$$

where  $\text{sg}$  denotes stop-gradient. Replay sampling supplies the outer transition mixture in Eq. (14). Thus PA-BCFM modifies only critic supervision. The policy proposal, action evaluation, and target-network update remain those of the matched flow-critic implementation.

The key point is that posterior averaging changes only the target noise, not the target mean. The following result holds for the practical finite pool.

**Theorem 1** (Finite-pool Rao–Blackwellization). *Assume that the bridge normalizer is positive and the endpoint direction has a finite conditional second moment. When  $K = 1$ ,  $\hat{u}_1 = u_I$ , so PA-BCFM exactly recovers BCFM. For every  $K \geq 1$ ,*

$$\begin{aligned}\hat{u}_K &= \mathbb{E}[u_I | X_t = x_t, \mathbf{Y}, \tau, t], \\ \mathbb{E}[\hat{u}_K | X_t = x_t, \tau, t] &= v_\tau^*(x_t, t).\end{aligned}\tag{25}$$

Moreover,

$$\text{Var}(\hat{u}_K | X_t, \tau, t) \leq \text{Var}(u_I | X_t, \tau, t).\tag{26}$$

Thus every finite pool preserves the population target and cannot increase conditional target variance.

Appendix B.2 gives the corresponding weighted total-variance equality. PA-BCFM removes posterior dispersion among compatible endpoint directions while leaving the conditional mean unchanged. The pool supplies additional information by conditioning away the sampled continuation identity, not by changing the Bellman endpoint law. Equivalently, for any prediction  $v$ ,

$$\mathbb{E}[(v - \hat{u}_K)^2 | X_t, \tau, t] = (v - v_\tau^*)^2 + \text{Var}(\hat{u}_K | X_t, \tau, t),\tag{27}$$

so every finite pool has the same population regression minimizer.

Under the moment conditions in Appendix B.3, the remaining large-pool variance obeys

$$\text{Var}(\hat{u}_K | x, t, \tau) = \frac{\sigma_\tau^2(x, t)}{K} + o(K^{-1}),\tag{28}$$

where

$$\sigma_\tau^2(x, t) = \frac{\mathbb{E}_{Y \sim \mu_\tau} [\ell_t(x | Y)^2 (u_t(x, Y) - v_\tau^*(x, t))^2]}{(\mathbb{E}_{Y \sim \mu_\tau} [\ell_t(x | Y)])^2}.\tag{29}$$

The effective sample size

$$\text{ESS}(\mathbf{w}) = \frac{1}{\sum_{j=1}^K w_j^2}\tag{30}$$

measures posterior concentration. Appendix B.3 shows that the normalized ESS converges to a transition-dependent concentration factor and controls the constant in the asymptotic variance bound. Larger pools help most when several candidates retain nonnegligible posterior mass; a larger nominal  $K$  cannot compensate for a fully concentrated posterior.

For the squared loss, if  $g_1$  is the BCFM gradient and  $g_K$  the PA-BCFM gradient at the same bridge point, then

$$\begin{aligned}g_K &= \mathbb{E}[g_1 | X_t = x_t, \mathbf{Y}, \tau, t], \\ \text{Cov}(g_K | X_t, \tau, t) &\preceq \text{Cov}(g_1 | X_t, \tau, t).\end{aligned}\tag{31}$$

Thus the same Rao–Blackwell ordering applies directly to target-induced gradient variability.

**Corollary 1** (Training and transport error). *Under the smoothness, Polyak–Łojasiewicz, and flow-Lipschitz assumptions in Appendix B.4, the pool-induced contribution to the SGD error floor is  $O((B_{\text{mb}}K)^{-1})$ , and its contribution to terminal  $W_1$  error is  $O((B_{\text{mb}}K)^{-1/2})$ . For a fixed policy, Bellman contraction amplifies the one-step error by at most  $1/(1 - \gamma)$ .*

The complete constants and the changing-policy tracking recursion are given in Appendix B.4.

### 3.3 Conservative Learning and Critic Diagnostics

Before each critic update, PA-BCFM has formed the weighted endpoint pool  $\{(Y_j, w_j)\}_{j=1}^K$ . This pool supports conservative reweighting and a training-time disagreement signal. Separate state–action diagnostics are computed directly from the twin return flows.

Let  $g : \mathbb{R} \rightarrow [0, \infty)$  favor the returns retained by a desired distortion. The reweighted posterior target is

$$w_j^g = \frac{w_j g(Y_j)}{\sum_{\ell=1}^K w_\ell g(Y_\ell)}, \quad (32)$$

$$\hat{u}_K^g = \sum_{j=1}^K w_j^g u_j. \quad (33)$$

The exact finite-pool Rao–Blackwell identity applies to the undistorted target  $g \equiv 1$ , but generally not to a distorted target  $g \neq 1$ . A nonincreasing  $g$  downweights high returns and yields a pessimistically distorted target; hard upper-tail truncation and the soft tilt  $g_\eta(y) = \exp(-\eta y)$  are two examples (Kuznetsov et al., 2020; Kumar et al., 2019, 2020; Ma et al., 2021). If the population distortion is  $L_g$ -Lipschitz in  $W_p$ , its Bellman operator is  $\gamma L_g$ -Lipschitz and is contractive when  $\gamma L_g < 1$ . Appendix B.5 gives the population definitions and proofs.

For  $c \in \{1, 2\}$ , the target critics share successor actions and return noises:

$$Y_j^{(c)} = r + \gamma(1 - d)\phi_{\bar{\theta}_c}(\epsilon'_j \mid 1, s', A'_j), \quad (34)$$

$$Y_j = \frac{1}{2}(Y_j^{(1)} + Y_j^{(2)}),$$

$$D_{\text{pool}}(\tau, x_t, t, \mathbf{Y}) = \sum_{j=1}^K w_j^0 (Y_j^{(1)} - Y_j^{(2)})^2. \quad (35)$$

Here  $A'_j \sim \pi(\cdot \mid s')$  and  $\epsilon'_j \sim \mathcal{N}(0, 1)$ , and  $w_j^0$  denotes the undistorted posterior weight obtained with  $g \equiv 1$ . Thus the statistic is defined before any adaptive reweighting. For task  $e$ , let  $\Delta_e = q_{.95}^{\mathcal{D}, e} - q_{.05}^{\mathcal{D}, e}$  be the fixed offline return-to-go range. The stop-gradient adaptive heuristic is

$$\begin{aligned} \eta(\tau, x_t, t) &= \eta_0 \operatorname{sg} \left[ \operatorname{clip} \left( \frac{\sqrt{D_{\text{pool}}}}{\Delta_e}, 0, 1 \right) \right], \\ g_{\tau, x_t, t}(y) &= \exp \left[ -\eta(\tau, x_t, t) \frac{y}{\Delta_e} \right]. \end{aligned} \quad (36)$$

For a fixed distortion  $g$ , Appendix B.5 establishes consistency and operator-level contraction conditions. The disagreement-adaptive schedule in Eq. (36) is a stop-gradient practical extension and is not covered by the fixed-distortion contraction result; its scope is clarified in Appendix B.6.

For a queried action, shared return noises isolate disagreement between the twin target return flows:

$$\begin{aligned} Z_m^{(c)}(s, a) &= \phi_{\bar{\theta}_c}(\xi_m \mid 1, s, a), \quad c \in \{1, 2\}, \\ \xi_m &\sim \mathcal{N}(0, 1), \quad m = 1, \dots, M_{\text{diag}}, \\ D_{\text{twin}}(s, a) &= \frac{1}{M_{\text{diag}}} \sum_{m=1}^{M_{\text{diag}}} (Z_m^{(1)}(s, a) - Z_m^{(2)}(s, a))^2. \end{aligned} \quad (37)$$

| Domain                                     | Gaussian Policies |        | Flow Policies |        | Distributional Critics |               |        |               |
|--------------------------------------------|-------------------|--------|---------------|--------|------------------------|---------------|--------|---------------|
|                                            | IQL               | ReBRAC | IFQL          | FQL    | IQN                    | Value Flows   | PCBF   | PA-BFM        |
| cube-double-play (5 tasks)                 | 6 ± 2             | 12 ± 3 | 14 ± 5        | 29 ± 6 | 42 ± 8                 | 69 ± 4        | 71 ± 5 | <b>79 ± 4</b> |
| cube-triple-play (5 tasks)                 | 1 ± 1             | 0 ± 0  | 0 ± 0         | 4 ± 2  | 6 ± 0                  | <b>14 ± 3</b> | 4 ± 1  | <b>14 ± 2</b> |
| puzzle-3x3-play (5 tasks)                  | 9 ± 3             | 22 ± 2 | 19 ± 1        | 30 ± 4 | 15 ± 1                 | 87 ± 13       | 53 ± 3 | <b>93 ± 3</b> |
| puzzle-4x4-play (5 tasks)                  | 7 ± 2             | 14 ± 3 | 25 ± 8        | 17 ± 5 | 27 ± 4                 | 27 ± 4        | 30 ± 4 | <b>52 ± 6</b> |
| scene-play (5 tasks)                       | 28 ± 3            | 41 ± 7 | 30 ± 4        | 56 ± 2 | 40 ± 1                 | 59 ± 4        | 54 ± 4 | <b>78 ± 2</b> |
| visual-antmaze-medium-navigate (5 tasks)   | 84 ± 5            | 87 ± 4 | 87 ± 2        | 38 ± 4 | 74 ± 4                 | 75 ± 10       | 75 ± 2 | <b>91 ± 3</b> |
| visual-antmaze-teleport-navigate (5 tasks) | 6 ± 4             | 4 ± 1  | 10 ± 4        | 5 ± 2  | 4 ± 2                  | 13 ± 4        | 14 ± 4 | <b>22 ± 5</b> |
| visual-cube-double-play (5 tasks)          | 11 ± 6            | 1 ± 1  | 2 ± 2         | 6 ± 1  | 1 ± 0                  | 13 ± 2        | 3 ± 1  | <b>17 ± 5</b> |
| visual-puzzle-3x3-play (5 tasks)           | 2 ± 3             | 20 ± 1 | 21 ± 0        | 20 ± 1 | 19 ± 1                 | <b>23 ± 2</b> | 18 ± 1 | 20 ± 1        |
| visual-scene-play (5 tasks)                | 26 ± 5            | 28 ± 5 | 21 ± 2        | 41 ± 4 | 41 ± 6                 | 43 ± 7        | 30 ± 3 | <b>50 ± 8</b> |

**Table 1:** Offline evaluation on OGBench. Each row averages five tasks; state-based and image-based results use eight and four seeds. Uncertainty is the arithmetic mean of the five task-level standard deviations. Matched flow critics share policy-side components while retaining their original critic supervision. Bold denotes the highest displayed mean in each domain.

For each critic, flow inversion gives

$$\hat{F}_c(z \mid s, a) = \Phi\left(\phi_{\theta_c}^{-1}(z \mid 1, s, a)\right), \quad (38)$$

$$\hat{F}_{\text{ens}}(z \mid s, a) = \frac{1}{2} \sum_{c=1}^2 \hat{F}_c(z \mid s, a),$$

$$P_{\text{out}}(s, a) = 1 - \hat{F}_{\text{ens}}(q_{\text{hi}}^{\mathcal{D}} \mid s, a). \quad (39)$$

$P_{\text{out}}$  is the twin-ensemble mass above an empirical upper-tail threshold fixed before diagnostic evaluation. Unlike these state-action diagnostics,  $D_{\text{pool}}$  is conditioned on a training transition and bridge point.

### 3.4 Distributional Policy Improvement

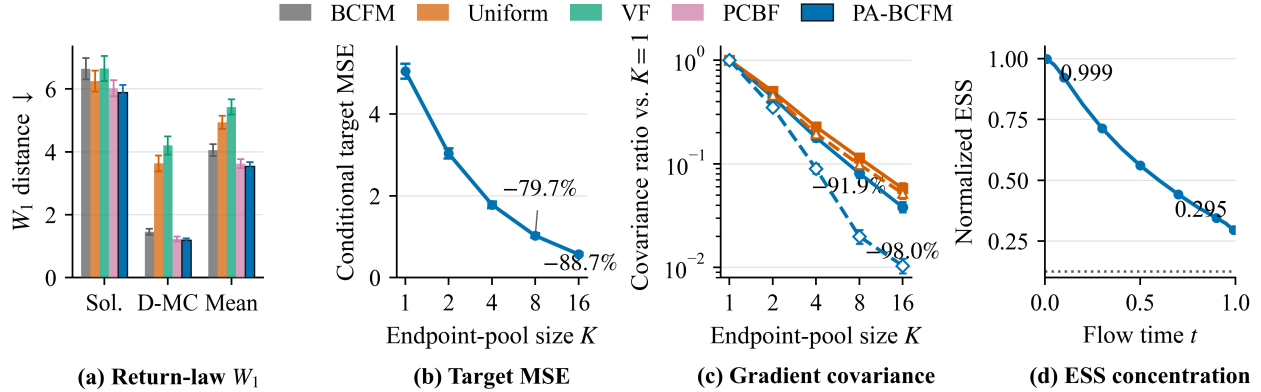
PA-BFM changes only the critic objective: a behavior flow proposes  $N$  actions, each evaluated with  $M_{\text{act}}$  shared return noises:

$$\begin{aligned} a_i &\sim \pi_{\beta}(\cdot \mid s), & i &= 1, \dots, N, \\ \epsilon_{im} &\sim \mathcal{N}(0, 1), & m &= 1, \dots, M_{\text{act}}, \\ z_{im}^{(c)} &= \phi_{\theta_c}(\epsilon_{im} \mid 1, s, a_i), & c &\in \{1, 2\}. \end{aligned} \quad (40)$$

The main experiments use a Monte Carlo estimate of mean return,

$$\begin{aligned} \hat{\mu}_i &= \frac{1}{2M_{\text{act}}} \sum_{c=1}^2 \sum_{m=1}^{M_{\text{act}}} z_{im}^{(c)}, \\ a^* &= \arg \max_{1 \leq i \leq N} \hat{\mu}_i. \end{aligned} \quad (41)$$

Appendix C.3 gives the shared candidate-selection and policy-training procedure. PA-BFM introduces no method-specific actor-side objective; matched flow-critic methods share the same policy-side procedure.



**Figure 1:** Controlled posterior-target evaluation. (a) Return-law  $W_1$ ; (b) conditional target MSE; (c) target-induced gradient trace-covariance ratio; and (d) normalized  $\text{ESS}(\mathbf{w})/K$  at  $K = 8$ . Sol./D-MC and VF denote Solitaire Dice/Discrete-MC and Value Flows; solid/dashed in (c) follow the same environment order, and dotted in (d) is  $1/K$ . Error bars are 95% confidence intervals; (b)–(c) use exact bridge-posterior anchors at fixed  $(\tau, t, x_t)$ .

## 4 Experiments

We separate estimator-level evaluation from downstream control. **(Q1)** Controlled processes test posterior-target error, gradient variability, and return-law fidelity. **(Q2)** Offline and offline-to-online benchmarks test downstream policy performance. **(Q3)** Ablations measure posterior weighting, pool size, and cost. **(Q4)** Diagnostics test reliability and pessimistic trade-offs.

The controlled suite comprises a Bernoulli-return process, Solitaire Dice, and a discrete Monte Carlo chain. Downstream evaluation covers 50 OGBench tasks and four offline-to-online scenarios (Park et al., 2025a). Baselines span scalar, flow-policy, and distributional critic families; RLPD appears only offline-to-online (Kostrikov et al., 2022; Tarasov et al., 2023; Park et al., 2025b; Bellemare et al., 2017; Dabney et al., 2018a; Ma et al., 2021; Dong et al., 2026; Xu et al., 2026; Ball et al., 2023).

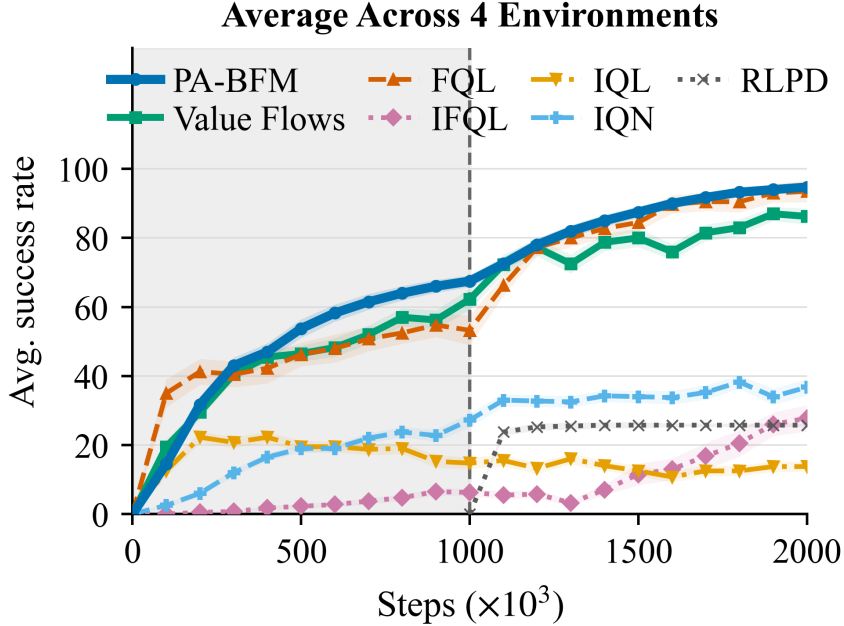
**Methods and evaluation.** We use a common JAX evaluation pipeline. BCFM, Value Flows, PCBF, and PA-BFM follow a matched code path and share policy-side and optimization components while retaining their critic objectives and flow-time supervision, isolating posterior averaging within this group. Hyperparameters are tuned once per domain under a common budget; state and image tasks use eight and four seeds.  $K = 8$  except in controlled and ablation studies. Full implementation, tuning, and evaluation details appear in Appendices D–E.

### 4.1 Controlled Posterior-Target Estimation

The controlled suite separates conditional target estimation from trained-critic fidelity. Bernoulli fixes the configuration, while Solitaire Dice and Discrete-MC are used for evaluation. For each fixed bridge query  $(\tau, t, x_t)$ , every replicate independently resamples the anchor from the exact bridge posterior and draws the remaining  $K - 1$  endpoints from  $\mu_\tau$ . We compare BCFM, uniform aggregation, and PA-BCFM for  $K \in \{1, 2, 4, 8, 16\}$ ; the learned-flow study includes Value Flows and PCBF with the fixed coefficient  $\lambda_{cv} = 0.3$  (Dong et al., 2026; Xu et al., 2026).

Figure 1(b–d) reports fixed-query estimates. Signed-bias 95% CIs contain zero for all  $K$ . Target MSE falls to 1.022 [0.97, 1.08] at  $K = 8$  and 0.570 [0.54, 0.61] at  $K = 16$ , reductions of 79.7%

and 88.7% relative to  $K = 1$ . At  $K = 8$ , the target-induced gradient trace-covariance ratios are 0.081 [0.073, 0.090] on Solitaire Dice and 0.020 [0.017, 0.023] on Discrete-MC. Normalized  $\text{ESS}(\mathbf{w})/K$  falls from 0.999 at  $t = 0.01$  to 0.295 at  $t = 0.99$ , indicating broad early averaging and sharper late selection. Figure 1(a) reports seed-level return-law fidelity: PA-BCFM has the lowest mean  $W_1$  on both processes and overall, 3.538 [3.41, 3.67]; its paired difference from PCBF is  $-0.085 [-0.156, -0.014]$ .



**Figure 2:** Offline-to-online evaluation. Mean success and one-standard-deviation bands over eight seeds on four OGBench scenarios. Gray marks one million offline optimizer updates, followed by one million online environment steps.

## 4.2 Offline and Online Performance

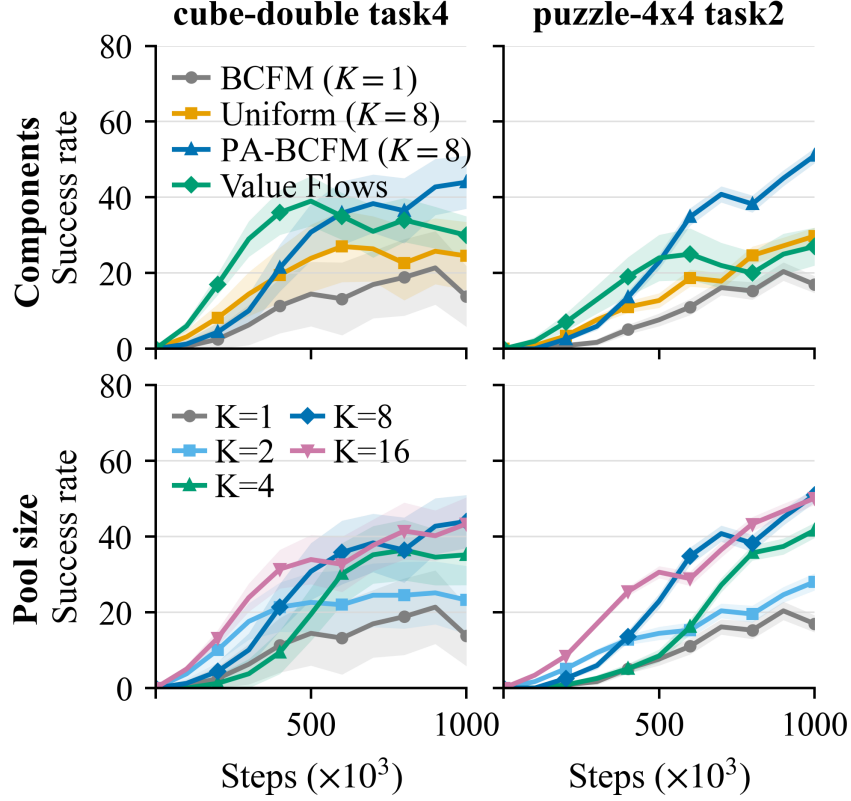
Table 1 summarizes offline performance. Fixed policy-side components isolate critic learning among matched flow critics. PA-BFM has the unique highest displayed mean on eight of ten domains and ties for the highest mean on one more. Relative to Value Flows, its state-based average rises from 51.2 to 63.2, its image-based average from 33.4 to 40.0, and the ten-domain average from 42.3 to 51.6. The task-level record is 35 wins, 7 ties, and 8 losses across both observation modalities. Full task-level results appear in Appendix Tables 7–8 of the supplement.

After one million offline optimizer updates and one million online environment steps, with an unchanged PA-BFM critic objective, Figure 2 shows mean aggregate success of 67.5 at online entry and a final mean of 94.75, versus 62.25 and 86.25 for Value Flows and 53.25 and 93.5 for FQL.

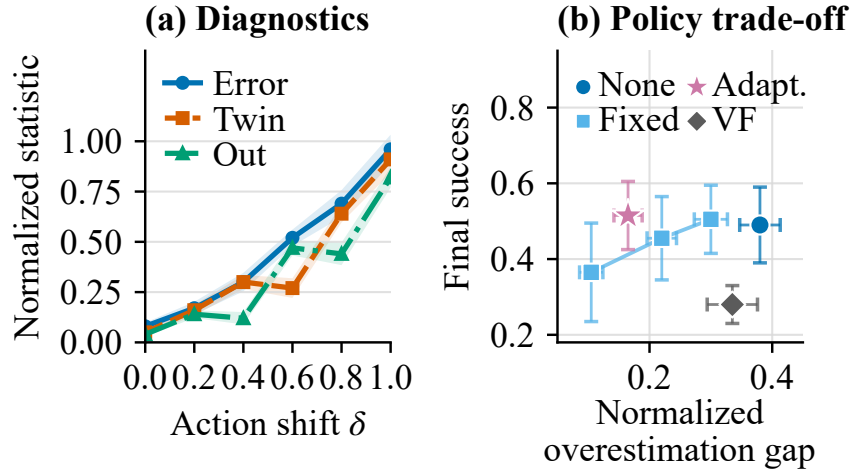
## 4.3 Estimator and Pool-Size Sensitivity

Figure 3 isolates posterior weighting under matched  $K = 8$  endpoints and training; only PA-BCFM weights endpoints by bridge compatibility. It improves final success by 19.5 and 21.2 points over uniform aggregation on `cube-double task4` and `puzzle-4x4 task2`, and by 14 and 24 points over Value Flows.

The pool-size sweep shows diminishing returns beyond  $K = 8$ : increasing  $K$  from 1 to 8 raises final success from 13.8 to 44 and from 17 to 51, whereas  $K = 16$  gives 43.4 and 50.1. On the



**Figure 3:** Estimator components and pool-size sensitivity. Top: BCFM ( $K = 1$ ), uniform and PA-BCFM ( $K = 8$ ), and Value Flows. Bottom: pool-size sensitivity on two state-based tasks.



**Figure 4:** Critic diagnostics and adaptive pessimism on `puzzle-4x4-play-task4`. Left: Error, Twin, and Out denote absolute prediction error, twin disagreement, and unsupported mass. Right: None, Fixed, Adapt., and VF denote undistorted, fixed, adaptive, and Value Flows variants.

representative state and visual configurations, the reported  $10^6$ -update costs at  $K = 8$  are 4.0 and 11.8 hours, respectively, below the corresponding Value Flows times of 4.4 and 12.5 hours. This performance-cost saturation motivates the default  $K = 8$ .

## 4.4 Critic Diagnostics and Adaptive Pessimism

On `puzzle-4x4-play-task4` at the final offline checkpoint, twin-critic disagreement and unsupported mass track absolute prediction error under action shift, with Spearman correlations  $\rho_{\text{twin}} = 0.943$  and  $\rho_{\text{out}} = 0.886$ . Fixed tilting reduces positive overestimation but eventually degrades success; the adaptive rule reduces it from 0.380 to 0.165 while raising final success from 0.490 to 0.515. Appendix D.4 provides the evaluation protocol and complete results for all evaluated fixed strengths.

## 5 Conclusion

We introduced PA-BFM, whose default target is an exact finite-pool Rao–Blackwellization of the single-endpoint BCFM direction under the exact latent-anchor posterior. It preserves the population velocity for every  $K$  and cannot increase conditional target variance. We derive  $K^{-1}$ -scaling variance and, under stated assumptions, optimization, terminal Wasserstein, and Bellman tracking bounds. The endpoint pool enables conservative reweighting, while the twin return flows provide critic-disagreement and unsupported-mass diagnostics. Controlled and OGBench experiments show improved estimation, return-law fidelity, and aggregate offline and offline-to-online performance over Value Flows.

## References

- Bhavya Agrawalla, Michal Nauman, Khush Agrawal, and Aviral Kumar. floq: Training critics via flow-matching for scaling compute in value-based rl, 2025.
- Michael S. Albergo and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023.
- 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*, pages 1577–1594, 2023.
- Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 449–458. PMLR, 2017.
- Deshu Chen, Yuchen Liu, Zhijian Zhou, Chao Qu, and Yuan Qi. Unleashing flow policies with distributional critics, 2025.
- Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1096–1105. PMLR, 2018a.
- Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, volume 32, 2018b.
- Shutong Ding, Yimiao Zhou, Ke Hu, Mokai Pan, Shan Zhong, Yanwei Fu, Jingya Wang, and Ye Shi. Distributional reinforcement learning with diffusion bridge critics, 2026.
- Perry Dong, Chongyi Zheng, Chelsea Finn, Dorsa Sadigh, and Benjamin Eysenbach. Value flows. In *International Conference on Learning Representations*, 2026.

- Jingliang Duan, Yang Guan, Shengbo Eben Li, Yangang Ren, Qi Sun, and Bo Cheng. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11):6584–6598, 2022. doi: 10.1109/TNNLS.2021.3082568.
- Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.
- 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*, volume 32, 2019.
- Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191, 2020.
- Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5556–5566. PMLR, 2020.
- Zeyang Li, Sunbochen Tang, and Navid Azizan. Reverse flow matching: A unified framework for online reinforcement learning with diffusion and flow policies. In *Proceedings of the 43rd International Conference on Machine Learning*. PMLR, 2026.
- 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.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2023.
- Yecheng Jason Ma, Dinesh Jayaraman, and Osbert Bastani. Conservative offline distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.
- 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*, pages 48104–48127, 2025b.
- George Stoica, Sayak Paul, Matthew Wallingford, Vivek Ramanujan, Abhay Nori, Winson Han, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Posterior-augmented flow matching, 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.
- Boyang Xu, Qing Zou, Siqin Yang, and Hao Yan. Path-coupled bellman flows for distributional reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning*. PMLR, 2026.
- Donglin Yang, Yongxing Zhang, Xin Yu, Liang Hou, Xin Tao, Pengfei Wan, Xiaojuan Qi, and Renjie Liao. Stable velocity: A variance perspective on flow matching, 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, 2025.

| Symbol              | Meaning                                                            | Symbol                   | Meaning                                                        |
|---------------------|--------------------------------------------------------------------|--------------------------|----------------------------------------------------------------|
| $\tau = (r, s', d)$ | transition outcome for fixed $(s, a)$                              | $\mu_\tau$               | conditional Bellman endpoint law                               |
| $Y, \mathbf{Y}$     | endpoint and pool $(Y_1, \dots, Y_K)$                              | $I$                      | uniformly sampled anchor index                                 |
| $X_t, x_t$          | bridge random variable and realization                             | $\epsilon, p_0$          | $\epsilon \sim \mathcal{N}(0, 1)$ and its density $p_0$        |
| $\ell_t(x   y)$     | bridge likelihood                                                  | $u_t(x, y)$              | endpoint-conditioned bridge direction                          |
| $u_j$               | candidate direction $u_t(x_t, Y_j)$                                | $w_j$                    | posterior anchor responsibility                                |
| $\hat{u}_K$         | posterior-mean training target                                     | $v_\theta$               | learned velocity field                                         |
| $v_\tau^*$          | transition-conditional population velocity                         | $\text{ESS}(\mathbf{w})$ | $1 / \sum_j w_j^2$                                             |
| $K$                 | ensemble-mean endpoint candidates $c \in \{1, 2\}$ in $\mathbf{Y}$ |                          | critic index                                                   |
| $j$                 | endpoint-candidate index, $1 \leq j \leq K$                        | $m$                      | action-evaluation sample index, $1 \leq m \leq M_{\text{act}}$ |

**Table 2:** Core notation. Uppercase variables denote random quantities and lowercase variables their realizations. Implementation- and proof-specific symbols are defined in the subsections where they are used.

## A Preliminaries

### A.1 Core Notation and Bellman Endpoint Law

Fix a state–action pair and transition outcome  $\tau = (r, s', d)$ , and let  $\mu_\tau$  be the conditional Bellman endpoint law. The estimator draws a pool whose twin-critic candidates are shared-noise ensemble means:

$$\mathbf{Y} = (Y_1, \dots, Y_K), \quad Y_j \stackrel{\text{i.i.d.}}{\sim} \mu_\tau, \quad (42)$$

$$Y_j^{(c)} = r + \gamma(1 - d)\phi_{\bar{\theta}_c}(\epsilon'_j | 1, s', A'_j), \quad c \in \{1, 2\},$$

$$Y_j = \frac{1}{2}(Y_j^{(1)} + Y_j^{(2)}). \quad (43)$$

Here  $A'_j \sim \pi(\cdot | s')$ ,  $\epsilon'_j \sim \mathcal{N}(0, 1)$ , and  $j = 1, \dots, K$ . Candidates are independent across  $j$ , both target critics share  $(A'_j, \epsilon'_j)$ , and  $\mu_\tau = \text{Law}(Y_j | \tau)$ .

### A.2 Distributional Bellman Operators

Let  $R$  and  $D$  denote reward and termination random variables. For a fixed policy  $\pi$ , the return satisfies

$$Z^\pi(s, a) \stackrel{d}{=} R + \gamma(1 - D)Z^\pi(S', A'), \quad (44)$$

$$A' \sim \pi(\cdot | S').$$

Under standard finite-moment assumptions, the induced operator is a  $\gamma$ -contraction in maximal Wasserstein distance (Bellemare et al., 2017). For  $d = 1$ , the conditional endpoint law is the point mass at  $r$ .

For an approximate iteration  $Z_{n+1} = \hat{\mathcal{T}}_n Z_n$ , we write  $\varepsilon_{K,n} = \bar{W}_1(\hat{\mathcal{T}}_n Z_n, \mathcal{T}^\pi Z_n)$ ; Appendix B.4 propagates this error.

### A.3 One-Dimensional Return Flows

A conditional return flow solves

$$\begin{aligned}\frac{d}{dt}\phi_\theta(\epsilon \mid t, s, a) &= v_\theta(\phi_\theta(\epsilon \mid t, s, a), t, s, a), \\ \phi_\theta(\epsilon \mid 0, s, a) &= \epsilon.\end{aligned}\tag{45}$$

In one dimension, uniqueness of the ODE makes the terminal map monotone and invertible on its image, enabling the inverse-flow CDF used in Appendix B.6.

## B Theoretical Analysis

This appendix proves the exact finite-pool Rao–Blackwell identity and its variance, optimization, transport, and Bellman-tracking implications. Each estimator analysis fixes the transition, target policy, and target critic, and uses  $t \in [0, 1 - \delta]$  for  $\delta > 0$ .

### B.1 Path Geometry and Population Bridge Target

Fix  $(s, a)$ , let  $(R, S', D)$  be the transition variables with realization  $\tau = (r, s', d)$ , and set  $X_0 = \epsilon \sim \mathcal{N}(0, 1)$ . The endpoint law  $\mu_\tau$  is defined in Eq. (7). The results condition on this law and hold for any fixed endpoint law satisfying the stated moments.

**Slice-wise Bellman geometry.** For a nonterminal transition ( $d = 0$ ), let the successor path be  $X'_t = (1 - t)X_0 + tX'_1$  with density  $p_t(\cdot \mid S', A')$  and velocity  $v_t(\cdot \mid S', A')$ , define the slice-wise affine Bellman path  $X_t^{\text{sl}} = r + \gamma X'_t$ . Conditional on  $R = r$ , the change of variables  $z = r + \gamma\xi$  gives the mixed density and velocity

$$q_t^{\text{slice}}(z) = \frac{1}{\gamma} \mathbb{E}_{S', A'} [p_t(\xi \mid S', A')], \quad \xi = \frac{z - r}{\gamma}, \tag{46}$$

$$u_t^{\text{slice}}(z) = \gamma \frac{\mathbb{E}[p_t(\xi \mid S', A') v_t(\xi \mid S', A')]}{\mathbb{E}[p_t(\xi \mid S', A')]}.\tag{47}$$

The second identity follows because each branch has  $\dot{X}_t^{\text{sl}} = \gamma \dot{X}'_t$ , so its numerator is the mixed probability flux.

At  $t = 0$ , however,  $X_0^{\text{sl}} = r + \gamma X_0 \sim \mathcal{N}(r, \gamma^2)$ , so the slice-wise path does not satisfy the common Gaussian source used by the critic. For a terminal transition,  $Y = r$  and  $\mu_\tau = \delta_r$ .

**Source-consistent endpoint bridge.** The bridge used by BCFM and PA-BCFM instead draws  $Y = r + \gamma(1 - d)X'_1$  and sets

$$\begin{aligned}X_t &= (1 - t)X_0 + tY, & X_0 &\perp Y, \\ X_0 &\sim \mathcal{N}(0, 1), & Y &\sim \mu_\tau.\end{aligned}\tag{48}$$

Using the same source noise in the nonterminal successor path gives

$$X_t = (1 - t)X_0 + t(R + \gamma X'_1) \tag{49}$$

$$= R + \gamma X'_t + (1 - t)[(1 - \gamma)X_0 - R], \tag{50}$$

which proves Eq. (11). The residual vanishes at  $t = 1$ , but not generally at intermediate times; thus the two paths share the Bellman endpoint without sharing their intermediate marginals or velocity fields.

Independence of  $X_0$  and  $Y \sim \mu_\tau$  gives

$$p(X_t = x, Y \in dy \mid \tau) = \ell_t(x \mid y) \mu_\tau(dy). \quad (51)$$

Bayes' rule therefore yields

$$\begin{aligned} \nu_{\tau,x}(dy) &= \frac{\ell_t(x \mid y) \mu_\tau(dy)}{B_\tau(x, t)}, \\ v_\tau^*(x, t) &= \int u_t(x, y) \nu_{\tau,x}(dy). \end{aligned} \quad (52)$$

**Lemma 1** (Population target of BCFM). *Fix  $(\tau, x, t)$  with  $t < 1$ . For the source-consistent bridge in Eq. (48), the conditional BCFM risk*

$$\mathcal{R}_\tau(u; x, t) = \mathbb{E} \left[ (u - (Y - X_0))^2 \mid X_t = x, \tau \right] \quad (53)$$

has the unique  $L_2$  minimizer

$$u^* = v_\tau^*(x, t) = \mathbb{E} \left[ \frac{Y - x}{1 - t} \mid X_t = x, \tau \right]. \quad (54)$$

*Proof.* Since  $X_t = (1 - t)X_0 + tY$ , conditioning on  $X_t = x$  gives

$$Y - X_0 = \frac{Y - x}{1 - t} = u_t(x, Y). \quad (55)$$

Let  $m_\tau(x, t) = \mathbb{E}[u_t(x, Y) \mid X_t = x, \tau]$ . The conditional squared risk decomposes as

$$\mathcal{R}_\tau(u; x, t) = \mathbb{E} \left[ (u - m_\tau + m_\tau - u_t(x, Y))^2 \mid X_t = x, \tau \right] \quad (56)$$

$$= (u - m_\tau(x, t))^2 \quad (57)$$

$$+ \text{Var} [u_t(x, Y) \mid X_t = x, \tau]. \quad (58)$$

The second term is independent of  $u$ , which proves the result.  $\square$

**Transition mixture.** Averaging  $v_\tau^*(x, t)$  under the posterior transition weight proportional to  $B_\tau(x, t)p(\tau \mid s, a)$  recovers the source-consistent population bridge velocity. Endpoint pooling therefore estimates the transition-conditional velocity, while replay sampling performs the outer Bellman mixture.

### Degeneracy of practical DCFM.

**Remark 1.** The practical DCFM residual admits degenerate shared fields. For example,  $v_\theta(z, t; s, a) = v_{\bar{\theta}}(z, t; s, a) = c(t)$  makes the residual vanish independently of the reward and transition. Thus the residual alone does not identify the Bellman transform; this observation does not assert that every optimization trajectory converges to a constant field.

## B.2 Finite-Pool Rao–Blackwellization

Fix a transition outcome  $\tau$  and a flow time  $t < 1$ . Draw

$$\begin{aligned} \mathbf{Y} &= (Y_1, \dots, Y_K), & Y_j &\stackrel{\text{i.i.d.}}{\sim} \mu_\tau, \\ I &\sim \text{Unif}\{1, \dots, K\}, & X_t &= (1-t)\epsilon + tY_I. \end{aligned} \quad (59)$$

where  $I$  is independent of the endpoint values and  $\epsilon \sim \mathcal{N}(0, 1)$ . For a realized bridge point  $x$ , define

$$\begin{aligned} u_j(x, t) &= \frac{Y_j - x}{1-t}, & \omega_j(x, t) &= \ell_t(x \mid Y_j), \\ w_j(x, t) &= \frac{\omega_j(x, t)}{\sum_{q=1}^K \omega_q(x, t)}. \end{aligned} \quad (60)$$

The single-endpoint BCFM target is  $U_I = u_I(X_t, t) = Y_I - \epsilon$ , whereas PA-BCFM uses

$$\hat{u}_K(X_t, t) = \sum_{j=1}^K w_j(X_t, t) u_j(X_t, t). \quad (61)$$

**Posterior anchor index.** Conditioned on  $(X_t = x, \mathbf{Y}, \tau, t)$ , the density of  $x$  under the hypothesis  $I = j$  is  $\omega_j(x, t)$ . Since the prior over  $I$  is uniform, Bayes' rule gives

$$\Pr(I = j \mid X_t = x, \mathbf{Y}, \tau, t) = w_j(x, t). \quad (62)$$

Consequently,

$$\hat{u}_K = \mathbb{E}[U_I \mid X_t, \mathbf{Y}, \tau, t]. \quad (63)$$

Taking the conditional expectation once more and using Lemma 1 yields

$$\begin{aligned} \mathbb{E}[\hat{u}_K \mid X_t, \tau, t] &= \mathbb{E}[U_I \mid X_t, \tau, t] \\ &= v_\tau^*(X_t, t), \quad \text{for every } K \geq 1. \end{aligned} \quad (64)$$

At  $K = 1$ ,  $\hat{u}_1 = U_I = Y_I - \epsilon$ , proving exact reduction to BCFM.

The outer transition mixture also gives  $\mathbb{E}[\hat{u}_K \mid X_t, s, a, t] = v^*(X_t, t \mid s, a)$ . Hence, for any square-integrable prediction  $u(X_t, s, a, t)$ ,

$$\mathbb{E}[(u - \hat{u}_K)^2] = \mathbb{E}[(u - v^*)^2] + \mathbb{E}[(\hat{u}_K - v^*)^2]. \quad (65)$$

The second term does not depend on  $u$ , so every finite pool preserves the population minimizer.

**Exact finite-pool variance reduction.** Applying conditional total variance to Eq. (63) gives

$$\begin{aligned} \text{Var}(U_I \mid X_t, \tau, t) &= \text{Var}(\hat{u}_K \mid X_t, \tau, t) \\ &\quad + \mathbb{E} \left[ \sum_{j=1}^K w_j (u_j - \hat{u}_K)^2 \mid X_t, \tau, t \right]. \end{aligned} \quad (66)$$

Therefore

$$\text{Var}(\hat{u}_K \mid X_t, \tau, t) \leq \text{Var}(U_I \mid X_t, \tau, t) \quad (67)$$

for every finite  $K$ . The reduction is strict whenever the posterior over anchor indices assigns positive mass to endpoints with different velocities.

**Relation to leave-one-out SNIS.** For asymptotic analysis, condition on the selected anchor and relabel it as  $Y_0$ . Then  $Y_0 \sim \nu_{\tau,x}$ , while  $Y_1, \dots, Y_{K-1}$  are i.i.d. from  $\mu_\tau$ . Define

$$\tilde{u}_{K-1} = \frac{\sum_{j=1}^{K-1} \omega_j u_j}{\sum_{j=1}^{K-1} \omega_j}. \quad (68)$$

Direct algebra yields

$$\begin{aligned} \hat{u}_K - \tilde{u}_{K-1} &= \alpha_{0,K}(u_0 - \tilde{u}_{K-1}), \\ \alpha_{0,K} &= \frac{\omega_0}{\sum_{j=0}^{K-1} \omega_j}. \end{aligned} \quad (69)$$

Under a positive normalizer and the stated moment conditions,  $\alpha_{0,K} = O_p(K^{-1})$ , so

$$\hat{u}_K - \tilde{u}_{K-1} = O_p(K^{-1}). \quad (70)$$

The practical estimator is therefore consistent and has the same first-order asymptotic distribution as standard SNIS. In finite samples, however, Eq. (64) is the sharper statement: the anchor correction exactly cancels the ratio bias in expectation under the uniform anchor-sampling protocol.

### B.3 Asymptotic Variance, ESS, and Gradient Covariance

**Ratio asymptotics.** Fix  $(x, t, \tau)$ , and write  $\omega(Y) = \ell_t(x | Y)$ ,  $u(Y) = u_t(x, Y)$ ,  $B_\tau(x, t) = \mathbb{E}_{\mu_\tau}[\omega(Y)]$ , and  $v_\tau^* = \mathbb{E}[\omega(Y)u(Y)]/B_\tau(x, t)$ . The multivariate central limit theorem for  $(\omega(Y)u(Y), \omega(Y))$ , followed by the delta method, gives influence variable

$$\psi(Y) = \frac{\omega(Y)(u(Y) - v_\tau^*)}{B_\tau(x, t)} \quad (71)$$

and asymptotic variance

$$\sigma_\tau^2(x, t) = \frac{\mathbb{E}_{\mu_\tau}[\omega(Y)^2(u(Y) - v_\tau^*)^2]}{(\mathbb{E}_{\mu_\tau}[\omega(Y)])^2}. \quad (72)$$

Because the anchor correction in Eq. (70) is  $o_p(K^{-1/2})$ , Slutsky's theorem gives

$$\begin{aligned} \sqrt{K}(\hat{u}_K - v_\tau^*) &\Rightarrow \mathcal{N}(0, \sigma_\tau^2(x, t)), \\ \text{Var}(\hat{u}_K | x, t, \tau) &= \frac{\sigma_\tau^2(x, t)}{K} + o(K^{-1}). \end{aligned} \quad (73)$$

**Effective sample size.** The realized effective sample size is

$$\text{ESS}(\mathbf{w}) = \frac{(\sum_j \omega_j)^2}{\sum_j \omega_j^2} = \frac{1}{\sum_j w_j^2}. \quad (74)$$

The anchor contributes only one term, so the strong law applied to the nonanchor endpoints yields

$$\frac{\text{ESS}(\mathbf{w})}{K} \xrightarrow{a.s.} \chi_\tau(x, t) = \frac{(\mathbb{E}_{\mu_\tau}[\omega(Y)])^2}{\mathbb{E}_{\mu_\tau}[\omega(Y)^2]}. \quad (75)$$

If  $|u(Y) - v_\tau^*| \leq C_{u,\tau}(x, t)$ , then

$$\begin{aligned} \sigma_\tau^2(x, t) &\leq \frac{C_{u,\tau}(x, t)^2}{\chi_\tau(x, t)}, \\ \text{Var}(\hat{u}_K | x, t, \tau) &\leq \frac{C_{u,\tau}(x, t)^2}{K \chi_\tau(x, t)} + o(K^{-1}). \end{aligned} \quad (76)$$

Thus ESS measures posterior-weight concentration and controls an asymptotic variance bound; it is not an unconditional exact variance identity.

**Gradient Rao–Blackwellization.** Fix  $(X_t, \tau, t, \theta)$ , and let  $J_\theta = \nabla_\theta v_\theta(X_t, t; s, a)$ . The single-endpoint BCFM gradient and PA-BCFM gradient are

$$g_1 = 2J_\theta(v_\theta - U_I), \quad g_K = 2J_\theta(v_\theta - \hat{u}_K). \quad (77)$$

Equation (63) implies

$$\begin{aligned} g_K &= \mathbb{E}[g_1 \mid X_t, \mathbf{Y}, \tau, t, \theta], \\ \text{Cov}(g_K \mid X_t, \tau, t, \theta) &\preceq \text{Cov}(g_1 \mid X_t, \tau, t, \theta). \end{aligned} \quad (78)$$

Moreover,

$$\text{Cov}(g_K \mid x, t, \tau, \theta) = \frac{4\sigma_\tau^2(x, t)}{K} J_\theta J_\theta^\top + o(K^{-1}). \quad (79)$$

This result requires no conditional independence assumption among candidate gradients.

## B.4 Optimization, Transport, and Bellman Tracking

**Relation to PCBF.** PCBF adds a learned successor-field control variate with coefficient  $\lambda_{\text{cv}}$ , so its population field differs from the source-consistent target by approximation and cross-conditioning terms (Xu et al., 2026). PA-BCFM instead preserves the exact population target for every finite  $K$ ; its pool-dependent error is variance rather than target bias.

**Optimization effect of the endpoint pool.** Define the population field risk

$$\begin{aligned} \mathcal{F}(\theta) &= \frac{1}{2} \mathbb{E} \left[ |v_\theta(X_t, t; s, a) - v^\star(X_t, t; s, a)|^2 \right], \\ \mathcal{F}_\star &= \inf_\theta \mathcal{F}(\theta). \end{aligned} \quad (80)$$

Exact finite-pool unbiasedness implies that the PA-BCFM stochastic gradient is unbiased for  $\nabla \mathcal{F}$ . Under uniform integrability of the conditional variance bound in Appendix B.3, total variance yields constants  $\sigma_{\text{base}}^2$  and  $C_{\text{pool}}$  such that a minibatch of size  $B_{\text{mb}}$  satisfies

$$\mathbb{E} \left[ \|\hat{g}_{K, B_{\text{mb}}} - \nabla \mathcal{F}(\theta)\|^2 \right] \leq \frac{1}{B_{\text{mb}}} \left( \sigma_{\text{base}}^2 + \frac{C_{\text{pool}}}{K} \right). \quad (81)$$

The first term collects transition, state–action, and flow-time randomness; the second is the endpoint-pool contribution.

Assume  $\mathcal{F}$  is  $L_{\mathcal{F}}$ -smooth and satisfies the Polyak–Łojasiewicz condition with constant  $\mu_{\text{PL}}$ , using  $\frac{1}{2} \|\nabla \mathcal{F}(\theta)\|_2^2 \geq \mu_{\text{PL}}(\mathcal{F}(\theta) - \mathcal{F}_\star)$ . For  $\theta_{n+1} = \theta_n - \eta_{\text{opt}} \hat{g}_{K, B_{\text{mb}}}$  with  $\eta_{\text{opt}} \leq 1/L_{\mathcal{F}}$ , smoothness and unbiasedness give

$$\begin{aligned} \mathbb{E}[\mathcal{F}(\theta_{n+1}) - \mathcal{F}_\star \mid \theta_n] &\leq (1 - \mu_{\text{PL}}\eta_{\text{opt}})(\mathcal{F}(\theta_n) - \mathcal{F}_\star) \\ &\quad + \frac{L_{\mathcal{F}}\eta_{\text{opt}}^2}{2B_{\text{mb}}} \left( \sigma_{\text{base}}^2 + \frac{C_{\text{pool}}}{K} \right). \end{aligned} \quad (82)$$

Unrolling the recursion proves

$$\begin{aligned} \mathbb{E}[\mathcal{F}(\theta_T) - \mathcal{F}_\star] &\leq (1 - \mu_{\text{PL}}\eta_{\text{opt}})^T (\mathcal{F}(\theta_0) - \mathcal{F}_\star) \\ &\quad + \frac{L_{\mathcal{F}}\eta_{\text{opt}}}{2\mu_{\text{PL}}B_{\text{mb}}} \left( \sigma_{\text{base}}^2 + \frac{C_{\text{pool}}}{K} \right), \end{aligned} \quad (83)$$

which gives the optimization statement in Corollary 1.

**Velocity-to-endpoint stability.** Couple the ideal and learned ODEs by their common source and let  $e_t = \hat{X}_t - X_t$ . If the learned field is  $L_v$ -Lipschitz in its state argument, then Grönwall’s inequality gives

$$|e_1| \leq \int_0^1 e^{L_v(1-t)} |v_{\theta_T}(X_t, t) - v^*(X_t, t)| dt. \quad (84)$$

Let  $Z_{\theta_T,1}^{\text{exact}}$  denote the terminal law obtained by exact integration of the learned ODE and  $Z_{\theta_T,1}^{(L)}$  the law produced by the  $L$ -step numerical solver. We write

$$\delta_{\text{ode}} = \mathbb{E}\left[W_1\left(Z_{\theta_T,1}^{(L)}, Z_{\theta_T,1}^{\text{exact}}\right)\right] \quad (85)$$

for the expected numerical-integration contribution.

Cauchy–Schwarz, the common-source coupling, Jensen’s inequality, and the triangle inequality imply

$$\begin{aligned} \mathbb{E}W_1(Z_{\theta_T,1}^{(L)}, Z_1^*) &\leq C_{L_v} \sqrt{2\mathbb{E}\mathcal{F}(\theta_T)} + \delta_{\text{ode}}, \\ C_{L_v}^2 &= \int_0^1 e^{2L_v(1-t)} dt. \end{aligned} \quad (86)$$

Substituting Eq. (83) into Eq. (86) proves the transport statement in Corollary 1. Thus the endpoint-pool contribution is  $O(K^{-1})$  in the optimization floor and  $O(K^{-1/2})$  in the terminal Wasserstein bound.

**Bellman tracking.** Let  $\varepsilon_{K,n}$  denote the one-step terminal-distribution error, including Eq. (86), numerical integration error, and any target-law mismatch. Distributional Bellman contraction gives, for fixed policy,

$$d_{n+1} \leq \gamma d_n + \varepsilon_{K,n}, \quad \limsup_n d_n \leq \frac{\limsup_n \varepsilon_{K,n}}{1 - \gamma}. \quad (87)$$

For a changing policy, one additional triangle inequality yields

$$d_{n+1} \leq \gamma d_n + \varepsilon_{K,n} + \Delta_n^\pi, \quad (88)$$

where  $\Delta_n^\pi$  measures the change between consecutive policy return laws. This is a tracking result, not a global actor–critic convergence claim.

## B.5 Conservative Endpoint Reweighting

Let  $g : \mathbb{R} \rightarrow [0, \infty)$  be a fixed distortion with  $0 < \int g(y) \mu_\tau(dy) < \infty$ . The distorted endpoint law is

$$\mu_{\tau,g}(dy) = \frac{g(y) \mu_\tau(dy)}{\int g(\tilde{y}) \mu_\tau(d\tilde{y})}. \quad (89)$$

After observing  $X_t = x$ , its bridge posterior is proportional to  $\ell_g(x | y) \mu_\tau(dy)$ , where

$$\ell_g(x | y) = \ell_t(x | y) g(y). \quad (90)$$

For  $g \equiv 1$ , the weights coincide with the posterior anchor probabilities in Eq. (62), giving the exact finite-pool Rao–Blackwell identity. For nonconstant  $g$ , the weights intentionally define a different target and are not the posterior of the uniformly selected anchor.

The practical distorted weights and target are

$$w_j^g = \frac{\ell_t(x | Y_j) g(Y_j)}{\sum_{\ell=1}^K \ell_t(x | Y_\ell) g(Y_\ell)}, \quad \hat{u}_K^g = \sum_{j=1}^K w_j^g u_t(x, Y_j). \quad (91)$$

**Proposition 1** (Consistency of the distorted endpoint estimator). *Under the moment and positive-normalizer assumptions of Theorem 1, condition on the selected anchor and relabel it as  $Y_0$ . With unnormalized weight  $\ell_g$ ,*

$$\begin{aligned}\hat{u}_K^g - \tilde{u}_{K-1}^g &= \alpha_{0,K}^g (u_0 - \tilde{u}_{K-1}^g), \\ \alpha_{0,K}^g &= \frac{\ell_g(x | Y_0)}{\sum_{j=0}^{K-1} \ell_g(x | Y_j)}.\end{aligned}\tag{92}$$

Consequently,  $\hat{u}_K^g - \tilde{u}_{K-1}^g = O_p(K^{-1})$ ,

$$\hat{u}_K^g \xrightarrow{p} v_{\tau,g}^*(x, t) = \frac{\mathbb{E}_{\mu_\tau}[\ell_g(x | Y) u_t(x, Y)]}{\mathbb{E}_{\mu_\tau}[\ell_g(x | Y)]},\tag{93}$$

and the practical estimator has the same first-order SNIS variance as the leave-one-out ratio. For nonconstant  $g$ , its finite-pool ratio is generally biased by  $O(K^{-1})$ ; the exact unbiasedness in Eq. (64) is specific to the undistorted objective.

*Proof.* Equation (92) follows from the same ratio identity as Eq. (69). The nonanchor samples remain independent draws from  $\mu_\tau$ , so the strong law, ratio expansion, and central limit theorem apply to  $\ell_g$ . The anchor contributes one additional term and changes the leave-one-out ratio by only  $O_p(K^{-1})$ . Unlike the case  $g \equiv 1$ , no tower-property cancellation makes this distorted ratio exactly unbiased at finite  $K$ .  $\square$

**Hard truncation.** For a population truncation level  $\rho \in [0, 1)$ , let  $q_{1-\rho}$  be the  $(1 - \rho)$ -quantile of  $\mu_\tau$  and define  $g_\rho^{\text{hard}}(y) = \mathbf{1}\{y \leq q_{1-\rho}\}$ . Replacing the population quantile by the empirical pool quantile is consistent under continuity of the endpoint CDF at the truncation point. The soft tilt  $g_\eta(y) = \exp(-\eta y)$  is covered directly by Proposition 1.

**Proposition 2** (Distorted Bellman contraction and pessimistic order). *Let  $\mathcal{D}_g$  be the population distortion and suppose*

$$W_p(\mathcal{D}_g \mu, \mathcal{D}_g \nu) \leq L_g W_p(\mu, \nu).\tag{94}$$

Then

$$\bar{W}_p(\mathcal{T}_g^\pi Z_1, \mathcal{T}_g^\pi Z_2) \leq \gamma L_g \bar{W}_p(Z_1, Z_2).\tag{95}$$

Hence  $\gamma L_g < 1$  gives a unique fixed point. If  $g$  is nonincreasing, then  $\mathcal{D}_g \mu \preceq_{\text{FSD}} \mu$ . If the Bellman operator is monotone and both distorted and undistorted iterations converge, their fixed points satisfy  $Z_g^\pi \preceq_{\text{FSD}} Z^\pi$ .

*Proof.* The Wasserstein bound follows from Bellman affine scaling and the assumed Lipschitz property of  $\mathcal{D}_g$ . For nonincreasing  $g$ , the standard covariance inequality implies first-order stochastic dominance. Bellman monotonicity then preserves this order along both convergent iterations.  $\square$

## B.6 Adaptive Distortion and Reliability Diagnostics

**Scope of the adaptive schedule.** Proposition 2 applies to a fixed population distortion  $g$ . The practical adaptive rule instead uses  $D_{\text{pool}}$ , which depends on the training transition, bridge point, and sampled endpoint pool, and is treated with stop-gradient within each critic update. Freezing this realized coefficient for differentiation does not define a common distorted Bellman operator across endpoint laws or training iterations. We therefore use the adaptive schedule only as a practical extension and make no operator-level contraction claim for it. This training-time quantity is distinct from the state-action diagnostic  $D_{\text{twin}}(s, a)$ .

**Inverse-flow support probability.** For a one-dimensional ODE with unique solutions, each target terminal map  $\epsilon \mapsto \phi_{\bar{\theta}_c}(\epsilon \mid 1, s, a)$  is order preserving. For  $c \in \{1, 2\}$ , define

$$\begin{aligned}\hat{F}_c(z \mid s, a) &= \Phi\left(\phi_{\bar{\theta}_c}^{-1}(z \mid 1, s, a)\right), \\ \hat{F}_{\text{ens}}(z \mid s, a) &= \frac{1}{2} \sum_{c=1}^2 \hat{F}_c(z \mid s, a).\end{aligned}\tag{96}$$

For the fixed empirical upper-return threshold  $q_{\text{hi}}^{\mathcal{D}}$ , the reported unsupported mass is

$$P_{\text{out}}(s, a) = 1 - \hat{F}_{\text{ens}}(q_{\text{hi}}^{\mathcal{D}} \mid s, a).\tag{97}$$

This is a model-implied tail probability rather than an OOD certificate.

**Statistical and dynamical conditions.** All finite-pool, asymptotic, optimization, and ODE conclusions hold under the assumptions stated in their respective propositions; none imply global actor-critic convergence.

## C Practical Algorithm

This appendix gives the full critic update, the optional endpoint-pool extensions, and the policy-improvement procedure used in the experiments.

### C.1 PA-BCFM Critic Update

**Endpoint construction.** For each transition  $(s, a, r, s', d)$  in a minibatch of size  $B_{\text{mb}}$ , write  $\tau = (r, s', d)$  and draw

$$A'_j \sim \pi(\cdot \mid s'), \quad \epsilon'_j \sim \mathcal{N}(0, 1), \quad j = 1, \dots, K,\tag{98}$$

and generate common Bellman endpoints from the ensemble-mean target return flow,

$$\begin{aligned}Y_j^{(c)} &= r + \gamma(1 - d)\phi_{\bar{\theta}_c}(\epsilon'_j \mid 1, s', A'_j), \\ Y_j &= \frac{1}{2} \left( Y_j^{(1)} + Y_j^{(2)} \right), \quad c \in \{1, 2\}.\end{aligned}\tag{99}$$

The two target critics share successor actions and return-noise samples. Their endpoint outputs are averaged into the common endpoint  $Y_j$ , and the resulting PA-BCFM target is used for both critic losses.

Both target flows are evaluated vectorially on the  $K$  shared samples and reuse the successor-state encoding. We sample  $I \sim \text{Unif}\{1, \dots, K\}$ , independently of endpoint values, then draw  $\epsilon \sim \mathcal{N}(0, 1)$  and  $t \sim \mathcal{U}[0, 1 - \epsilon_t]$ . The bridge point is

$$x_t = (1 - t)\epsilon + tY_I.\tag{100}$$

**Posterior target.** For a distortion  $g$ , define the log weight

$$\Lambda_j = -\frac{(x_t - tY_j)^2}{2(1-t)^2} + \log g(Y_j). \quad (101)$$

The default PA-BCFM objective uses  $g \equiv 1$ . We normalize with log-sum-exp,

$$w_j = \exp\left(\Lambda_j - \text{logsumexp}_{\ell=1}^K \Lambda_\ell\right), \quad (102)$$

and construct the candidate directions and posterior target

$$u_j = \frac{Y_j - x_t}{1-t}, \quad \hat{u}_K = \sum_{j=1}^K w_j u_j. \quad (103)$$

The endpoints, weights, and posterior target are stop-gradient quantities.

**Critic and target updates.** Each member of the twin critic minimizes

$$\mathcal{L}_c = \mathbb{E} \left[ \|v_{\theta_c}(x_t, t; s, a) - \text{sg}[\hat{u}_K]\|_2^2 \right], \quad c \in \{1, 2\}. \quad (104)$$

The two online critics are parameterized and optimized separately despite sharing this target; their individual target flows are retained for the training-time pool signal and direct state–action diagnostics in Appendix C.2.

After the gradient update, the corresponding target parameters are updated by Polyak averaging,

$$\bar{\theta}_c \leftarrow \tau_{\text{tgt}} \theta_c + (1 - \tau_{\text{tgt}}) \bar{\theta}_c. \quad (105)$$

Gradients do not pass through the target flow, posterior construction, diagnostics, or target update.

## C.2 Conservative Reweighting and Diagnostics

**Fixed endpoint distortions.** The main PA-BCFM experiments use the undistorted target  $g \equiv 1$ . The reported conservative variant uses the fixed soft tilt

$$g_\eta(Y_j) = \exp(-\eta Y_j), \quad \eta \geq 0. \quad (106)$$

Hard upper-tail truncation is implemented by zeroing candidates above the empirical pool quantile; it is not used in the reported experiments. The exact finite-pool Rao–Blackwell identity applies to  $g \equiv 1$ ; nonconstant  $g$  deliberately defines a distorted target.

**Training-time pool disagreement.** To separate critic disagreement from candidate-sampling noise, both target critics are evaluated on the same successor actions and return noises:

$$Y_j^{(c)} = r + \gamma(1-d)\phi_{\bar{\theta}_c}(\epsilon'_j \mid 1, s', A'_j), \quad c \in \{1, 2\}. \quad (107)$$

Let  $Y_j = (Y_j^{(1)} + Y_j^{(2)})/2$ , and compute the undistorted posterior weights  $w_j^0$  from  $Y_j$ . The disagreement score is

$$D_{\text{pool}}(\tau, x_t, t, \mathbf{Y}) = \sum_{j=1}^K w_j^0 \left( Y_j^{(1)} - Y_j^{(2)} \right)^2. \quad (108)$$

---

**Algorithm 1** Posterior-Averaged Bellman Flow Matching (PA-BFM)

---

**Require:** replay buffer  $\mathcal{D}$ , twin critics and targets, policy  $\pi$ , behavior proposal  $\pi_\beta$ , endpoint count  $K$ , optional endpoint distortion  $g$ , policy-training schedule  $\mathcal{P}$

```
1: repeat
2:   Sample  $B_{\text{mb}}$  transitions  $(s, a, r, s', d)$  from  $\mathcal{D}$ 
3:   Draw  $K$  successor actions and target-flow noises per transition
4:   Evaluate both targets and set  $Y_j = (Y_j^{(1)} + Y_j^{(2)})/2$ 
5:   Sample  $I$  uniformly and construct  $x_t = (1 - t)\epsilon + tY_I$ 
6:   Form stable posterior logits, weights, and posterior target  $\hat{u}_K$ 
7:   Update both critics using Eq. (104)
8:   Polyak-update both target critics
9:   if online interaction is active then
10:    Collect transitions and append them to  $\mathcal{D}$ 
11:   end if
12:   if one-step policy training is enabled by  $\mathcal{P}$  then
13:    Update the behavior-regularized one-step policy with Eq. (114)
14:   end if
15: until the training budget is exhausted
16: return twin return critics, behavior proposal, and optional one-step policy
```

---

**Adaptive distortion.** For the reported diagnostic experiment, return-valued quantities are scaled by the task-specific  $\Delta_e = q_{.95}^{\mathcal{D},e} - q_{.05}^{\mathcal{D},e}$  return-to-go range, and normalized disagreement is clipped to  $[0, 1]$ . Training-time pool disagreement modulates the soft tilt as

$$\begin{aligned}\eta(\tau, x_t, t) &= \eta_0 \operatorname{sg} \left[ \operatorname{clip} \left( \frac{\sqrt{D_{\text{pool}}(\tau, x_t, t, \mathbf{Y})}}{\Delta_e}, 0, 1 \right) \right], \\ g_{\tau, x_t, t}(Y_j) &= \exp \left[ -\eta(\tau, x_t, t) \frac{Y_j}{\Delta_e} \right].\end{aligned}\tag{109}$$

The stop-gradient makes the diagnostic a target-construction signal rather than a second gradient path through the critic. This sample-dependent rule is not covered by the fixed-distortion contraction result.

**State-action reliability diagnostics.** At a queried  $(s, a)$ , shared Gaussian samples isolate disagreement between the twin target return flows:

$$\begin{aligned}Z_m^{(c)}(s, a) &= \phi_{\bar{\theta}_c}(\xi_m \mid 1, s, a), \quad c \in \{1, 2\}, \\ \xi_m &\sim \mathcal{N}(0, 1), \quad m = 1, \dots, M_{\text{diag}}, \\ D_{\text{twin}}(s, a) &= \frac{1}{M_{\text{diag}}} \sum_{m=1}^{M_{\text{diag}}} (Z_m^{(1)}(s, a) - Z_m^{(2)}(s, a))^2.\end{aligned}\tag{110}$$

For each one-dimensional return flow, inversion gives the critic CDF; the reported unsupported

mass uses their ensemble average:

$$\begin{aligned}\widehat{F}_c(b \mid s, a) &= \Phi\left(\phi_{\theta_c}^{-1}(b \mid 1, s, a)\right), \\ \widehat{F}_{\text{ens}}(b \mid s, a) &= \frac{1}{2} \sum_{c=1}^2 \widehat{F}_c(b \mid s, a), \\ P_{\text{out}}(s, a) &= 1 - \widehat{F}_{\text{ens}}(q_{\text{hi}}^{\mathcal{D}} \mid s, a).\end{aligned}\tag{111}$$

The threshold  $q_{\text{hi}}^{\mathcal{D}}$  is a dataset-derived upper-return threshold selected once before diagnostic evaluation and then held fixed.

### C.3 Policy Extraction and Shared Policy Training

All main experiments rank candidate actions by predicted mean return. Where policy training is enabled by  $\mathcal{P}$ , actions are distilled into a one-step policy using the shared behavior-regularized objective below. This policy-side procedure is matched across PA-BFM, Value Flows, and PCBF.

**Distributional action evaluation.** For a state  $s$ , the behavior flow proposes  $N$  actions  $a_i \sim \pi_\beta(\cdot \mid s)$ . Each action is evaluated with  $M_{\text{act}}$  shared return noises under the twin return flows:

$$\begin{aligned}z_{im}^{(c)} &= \phi_{\theta_c}(\epsilon_{im} \mid 1, s, a_i), \\ \epsilon_{im} &\sim \mathcal{N}(0, 1), \quad \begin{matrix} m=1, \dots, M_{\text{act}}, \\ c \in \{1, 2\} \end{matrix}.\end{aligned}\tag{112}$$

The mean score and selected action are

$$\begin{aligned}\widehat{\mu}_i &= \frac{1}{2M_{\text{act}}} \sum_{c=1}^2 \sum_{m=1}^{M_{\text{act}}} z_{im}^{(c)}, \\ a^* &= \arg \max_{1 \leq i \leq N} \widehat{\mu}_i.\end{aligned}\tag{113}$$

With the default  $M_{\text{act}} = 1$  in Table 12,  $\widehat{\mu}_i$  is a single-sample Monte Carlo estimate of the critic mean.

**Shared one-step policy update.** When policy training is enabled by  $\mathcal{P}$ , a one-step policy  $\pi_\psi$ , with parameters  $\psi$ , is trained against the frozen critic with behavior regularization:

$$\begin{aligned}a_\psi &= \pi_\psi(s, \zeta), \\ \mathcal{L}_{\text{policy}}(\psi) &= \mathbb{E}_{\substack{s \sim \mathcal{D} \\ \zeta \sim \mathcal{N}(0, \mathbf{I}_{d_a})}} \left[ -\widehat{\mu}(s, a_\psi) \right. \\ &\quad \left. + \alpha_{\text{BC}} \|a_\psi - \pi_\beta(s, \zeta)\|_2^2 \right].\end{aligned}\tag{114}$$

Here  $\widehat{\mu}(s, a)$  is the twin-flow sample mean above evaluated at action  $a$ ,  $\zeta$  is action-generation noise,  $\mathbf{I}_{d_a}$  is the identity covariance in action dimension, and  $\alpha_{\text{BC}} \geq 0$  controls behavior regularization. Critic parameters are fixed during this actor update.

### C.4 Implementation and Complexity

**Numerical implementation.** We restrict flow time to  $t \leq 1 - \epsilon_t$ , compute posterior weights with log-sum-exp, and reuse the successor-state representation across all  $K$  endpoints. Target-flow solves are vectorized and run without gradient storage. The same ODE solver and precision policy are used across estimator variants.

| Process          | Configuration                                                                                                                                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bernoulli return | One-state continuing MRP; $R \sim \text{Bernoulli}(0.5)$ ; $\gamma = 0.5$ ; exact return law $Z \sim \text{Unif}(0, 2)$ .                                                                                                                                                                                                            |
| Solitaire Dice   | A fair six-sided die is repeatedly rolled. Rolls 2–6 produce reward 1 and continue; roll 1 produces reward 0 and terminates; $\gamma = 0.9$ .                                                                                                                                                                                        |
| Discrete-MC      | Twenty-state nearest-neighbor chain with absorbing terminal states 0 and 19. States 1–18 use the fixed cosine–exponential nearest-neighbor transition weights of the controlled-process configuration. Transitions to nonterminal states produce reward 1, transitions into a terminal state produce reward 0, and $\gamma = 0.95$ . |

**Table 3:** Controlled-process settings. The Bernoulli process is used to fix the controlled-study configuration; Solitaire Dice and Discrete-MC are used for the reported evaluation.

**Complexity.** With  $T_{\text{flow}}$  target-flow solver steps, endpoint construction costs  $O(B_{\text{mb}}KT_{\text{flow}})$  network work and  $O(B_{\text{mb}}K)$  posterior arithmetic and storage. Policy extraction requires  $O(NM_{\text{act}}T_{\text{flow}})$  return-flow evaluations per state. End-to-end wall time is reported separately from these asymptotic counts.

## D Additional Experiments

### D.1 Controlled Estimator Studies

**Controlled processes and exact references.** We use a one-state Bernoulli-return process to fix the controlled-study configuration and evaluate the frozen configuration on Solitaire Dice and a discrete Monte Carlo chain. All three processes admit exact return laws, so the finite-pool posterior target can be evaluated without a learned target critic. Table 3 summarizes the process definitions.

For a controlled conditional endpoint law  $\mu$ , the exact posterior Bellman velocity at bridge point  $x$  and flow time  $t$  is

$$v^*(x, t) = \frac{\mathbb{E}_{Y \sim \mu} [\ell_t(x | Y) u_t(x, Y)]}{\mathbb{E}_{Y \sim \mu} [\ell_t(x | Y)]}, \quad (115)$$

where  $\ell_t(x | Y)$  is the Gaussian bridge likelihood. The expectations are evaluated from the exact endpoint laws. For each reported process, the oracle study uses  $N_q = 4096$  bridge queries and  $N_{\text{rep}} = 512$  endpoint-pool replicates per query. For query  $q$ , we sample and fix  $(\tau_q, t_q, x_q)$  from the exact bridge marginal and compute the oracle velocity  $v_q^*$ . In every replicate, the relabeled anchor is resampled from the exact bridge posterior

$$Y_{\text{anc}, q}^{(r)} \sim \nu_{\tau_q, x_q}(dy) \propto \ell_{t_q}(x_q | y) \mu_{\tau_q}(dy). \quad (116)$$

The remaining  $K - 1$  endpoints are sampled independently from  $\mu_{\tau_q}$ . This is the conditional law of the uniform-anchor estimator given the fixed bridge query, so the  $K = 1$  target remains random and provides a nonzero BCFM reference. Uniform aggregation and PA-BCFM use matched candidate pools within each comparison. Process-level results give equal weight to Solitaire Dice and Discrete-MC. Confidence intervals use a process-stratified query-cluster bootstrap, retaining all replicates of each resampled query together.

**Fixed-query conditional target estimation.** We evaluate the anchor-in-pool estimator with  $K \in \{1, 2, 4, 8, 16\}$ . For query  $q$  and endpoint-pool replicate  $r$ , let  $\hat{u}_{K, q}^{(r)}$  denote the finite-pool target.

| $K$ | Signed bias [95% CI]   | Cond. variance | MSE [95% CI]        |
|-----|------------------------|----------------|---------------------|
| 1   | 0.004 [−0.006, 0.014]  | 5.0407         | 5.0447 [4.86, 5.23] |
| 2   | −0.003 [−0.011, 0.005] | 3.0301         | 3.0331 [2.91, 3.16] |
| 4   | 0.002 [−0.005, 0.009]  | 1.7790         | 1.7810 [1.70, 1.86] |
| 8   | −0.001 [−0.006, 0.004] | 1.0209         | 1.0219 [0.97, 1.08] |
| 16  | 0.001 [−0.003, 0.005]  | 0.5693         | 0.5703 [0.54, 0.61] |

**Table 4:** Fixed-query conditional target estimation. Bias is the mean signed error; variance is the mean within-query conditional variance. Brackets give 95% process-stratified query-cluster bootstrap confidence intervals. Lower MSE is better.

| Process     | Estimator | $K = 1$                 | $K = 2$                     | $K = 4$                     | $K = 8$                     | $K = 16$                      |
|-------------|-----------|-------------------------|-----------------------------|-----------------------------|-----------------------------|-------------------------------|
| Solitaire   | Uniform   | 1.000<br>[1.000, 1.000] | .499<br>[.470, .529]        | .227<br>[.208, .247]        | .114<br>[.102, .126]        | .058<br>[.051, .065]          |
| Solitaire   | PA-BCFM   | 1.000<br>[1.000, 1.000] | <b>.438</b><br>[.409, .468] | <b>.179</b><br>[.164, .195] | <b>.081</b><br>[.073, .090] | <b>.038</b><br>[.034, .043]   |
| Discrete-MC | Uniform   | 1.000<br>[1.000, 1.000] | .460<br>[.432, .489]        | .195<br>[.179, .212]        | .098<br>[.089, .108]        | .052<br>[.046, .058]          |
| Discrete-MC | PA-BCFM   | 1.000<br>[1.000, 1.000] | <b>.349</b><br>[.322, .377] | <b>.090</b><br>[.080, .100] | <b>.020</b><br>[.017, .023] | <b>.010</b><br>[.0088, .0122] |

**Table 5:** Target-induced gradient trace-covariance ratio. Values are normalized by the corresponding random  $K = 1$  BCFM reference; brackets give 95% query-cluster bootstrap confidence intervals. Lower is better.

We report the conditional mean-squared error

$$\text{MSE}_K = \frac{1}{N_q N_{\text{rep}}} \sum_{q=1}^{N_q} \sum_{r=1}^{N_{\text{rep}}} \left( \hat{u}_{K,q}^{(r)} - v_q^* \right)^2. \quad (117)$$

**Gradient variability and posterior concentration.** For the gradient study, critic parameters and the bridge query  $(\tau, t, x_t)$  are fixed, while every replicate independently resamples the posterior anchor and the remaining endpoint pool using the same conditional protocol as above. Hence the  $K = 1$  target and gradient remain random. For gradient replicate  $g_r$ , we compute the trace covariance

$$\text{GVar} = \text{tr} \left( \widehat{\text{Cov}}[g] \right) = \frac{1}{R_g} \sum_{r=1}^{R_g} \|g_r - \bar{g}\|_2^2. \quad (118)$$

For each query, we compute the trace covariance across  $R_g = 256$  replicates. The reported process-level ratio divides the aggregate query-level covariance trace at pool size  $K$  by the corresponding  $K = 1$  aggregate. These ratios measure fixed-query, target-induced gradient variability rather than the unconditional variance of full SGD updates. Confidence intervals use the same query-cluster bootstrap.

For PA-BCFM, posterior concentration is summarized by the normalized effective sample size

$$\frac{\text{ESS}(\mathbf{w})}{K} = \frac{1}{K \sum_{j=1}^K w_j^2}. \quad (119)$$

ESS is used as a within-estimator measure of posterior concentration; cross-estimator variance is assessed directly by Table 5.

**Learned return-law fidelity.** Finally, we train BCFM, uniform endpoint aggregation, Value Flows, PCBF, and PA-BCFM using the same critic architecture, optimizer, transition batches, and

50,000-update budget. Each configuration is trained with eight seeds, and return-law fidelity is evaluated using 50,000 return samples per evaluated state–action pair. Solitaire Dice is evaluated before the first roll. For Discrete-MC, the process-level fidelity value aggregates the configured statewise evaluation grid.

| Method                        | Solitaire                   | Discrete-MC                 | Avg.                        |
|-------------------------------|-----------------------------|-----------------------------|-----------------------------|
| BCFM ( $K = 1$ )              | 6.644<br>[6.31,6.98]        | 1.454<br>[1.36,1.55]        | 4.049<br>[3.86,4.24]        |
| Uniform ( $K = 8$ )           | 6.252<br>[5.92,6.58]        | 3.630<br>[3.38,3.88]        | 4.941<br>[4.73,5.15]        |
| Value Flows                   | 6.650<br>[6.25,7.05]        | 4.202<br>[3.91,4.49]        | 5.426<br>[5.18,5.67]        |
| PCBF ( $\lambda_{cv} = 0.3$ ) | 6.020<br>[5.76,6.28]        | 1.227<br>[1.15,1.30]        | 3.623<br>[3.48,3.77]        |
| PA-BCFM ( $K = 8$ )           | <b>5.891</b><br>[5.65,6.13] | <b>1.185</b><br>[1.12,1.25] | <b>3.538</b><br>[3.41,3.67] |

**Table 6:** Learned return-law fidelity. Entries are seed-level means with 95% confidence intervals. Avg. is the equal-weight average over Solitaire Dice and Discrete-MC. Lower 1-Wasserstein distance is better.

## D.2 Full Offline OGBench Results

Tables 7 and 8 report the complete task-level results underlying Table 1. Seed counts and the domain-level tuning-task convention follow Appendix E.3.

## D.3 Offline-to-Online Results

Table 9 reports scenario-level performance at online entry and at the final online checkpoint.

## D.4 Critic Diagnostics and Adaptive Pessimism

We evaluate critic reliability at the final  $10^6$ -update checkpoint on `puzzle-4x4-play-task4` using eight seeds. This prespecified domain-level tuning task is not selected using the diagnostic results. For each of  $N_q = 128$  held-out state–action pairs, we construct

$$\begin{aligned} a_\delta &= \text{clip}((1 - \delta)a_{\mathcal{D}} + \delta a_{\text{rand}}, -1, 1), \\ \delta &\in \{0, .2, .4, .6, .8, 1\}. \end{aligned} \tag{120}$$

where  $a_{\text{rand}}$  is sampled uniformly from the normalized action space and shared across methods. Absolute prediction error compares the twin-critic mean with a Monte Carlo return averaged over  $N_{\text{ref}} = 64$  rollouts after forcing  $a_\delta$  as the first action and then following the checkpoint’s evaluation policy. The assessed critic is not used in this reference. We compute  $D_{\text{twin}}$  from  $M_{\text{diag}} = 64$  shared return-noise samples and define  $P_{\text{out}}$  as predicted mass above the 95th percentile of offline-dataset return-to-go. Prediction error,  $\sqrt{D_{\text{twin}}}$ , and positive overestimation are normalized by the difference between the 95th and 5th return-to-go percentiles; unsupported mass is already dimensionless. Metrics are averaged over queries and seeds.

Across the six action-shift levels, both diagnostics track absolute return-prediction error. The descriptive Spearman correlations are  $\rho_{\text{twin}} = 0.943$  for twin-critic disagreement and  $\rho_{\text{out}} = 0.886$  for unsupported mass.

For conservative training, we compare the undistorted target, fixed soft tilts  $\eta \in \{0.1, 0.3, 0.6\}$ , the adaptive rule in Eq. (36), and Value Flows. The adaptive rule uses  $D_{\text{pool}}$  computed from undistorted posterior weights, clips normalized disagreement to  $[0, 1]$ , and caps the tilt at  $\eta_0 = 0.6$ . It is a stop-gradient practical heuristic and is not directly covered by the fixed-distortion contraction

| Task                    | Gaussian Policies |        |         | Flow Policies |                | Distributional Critics |                |              |               |                |                |
|-------------------------|-------------------|--------|---------|---------------|----------------|------------------------|----------------|--------------|---------------|----------------|----------------|
|                         | BC                | IQL    | ReBRAC  | IFQL          | FQL            | C51                    | IQN            | CODAC        | Value Flows   | PCBF           | PA-BFM         |
| <i>cube-double-play</i> |                   |        |         |               |                |                        |                |              |               |                |                |
| task1                   | 8 ± 3             | 27 ± 5 | 45 ± 6  | 35 ± 9        | 61 ± 9         | 9 ± 0                  | 70 ± 14        | 80 ± 11      | 97 ± 1        | 92 ± 3         | <b>99 ± 1</b>  |
| task2*                  | 0 ± 0             | 1 ± 1  | 7 ± 3   | 9 ± 5         | 36 ± 6         | 0 ± 0                  | 24 ± 9         | 63 ± 4       | 76 ± 7        | 74 ± 7         | <b>89 ± 2</b>  |
| task3                   | 0 ± 0             | 0 ± 0  | 4 ± 1   | 8 ± 5         | 22 ± 5         | 0 ± 0                  | 25 ± 6         | 66 ± 9       | 73 ± 4        | 81 ± 8         | <b>91 ± 2</b>  |
| task4                   | 0 ± 0             | 0 ± 0  | 1 ± 1   | 1 ± 1         | 5 ± 2          | 0 ± 0                  | 10 ± 1         | 13 ± 2       | 30 ± 5        | 22 ± 5         | <b>44 ± 7</b>  |
| task5                   | 0 ± 0             | 4 ± 3  | 4 ± 2   | 17 ± 6        | 19 ± 10        | 0 ± 0                  | 81 ± 8         | 82 ± 4       | 69 ± 5        | <b>84 ± 3</b>  | 70 ± 6         |
| <i>cube-triple-play</i> |                   |        |         |               |                |                        |                |              |               |                |                |
| task1*                  | 1 ± 1             | 4 ± 4  | 1 ± 2   | 2 ± 2         | 20 ± 6         | 1 ± 0                  | 29 ± 2         | 9 ± 5        | 59 ± 12       | 18 ± 4         | <b>60 ± 8</b>  |
| task2                   | 0 ± 0             | 0 ± 0  | 0 ± 0   | 0 ± 0         | <b>1 ± 2</b>   | 0 ± 0                  | 0 ± 0          | <b>1 ± 0</b> | 0 ± 0         | 0 ± 1          | 0 ± 0          |
| task3                   | 0 ± 0             | 0 ± 0  | 0 ± 0   | 0 ± 0         | 0 ± 0          | 0 ± 0                  | 1 ± 0          | 0 ± 0        | <b>7 ± 3</b>  | 1 ± 1          | <b>7 ± 2</b>   |
| task4                   | 0 ± 0             | 0 ± 0  | 0 ± 0   | 0 ± 0         | 0 ± 0          | 0 ± 0                  | 0 ± 0          | 0 ± 0        | 0 ± 0         | 0 ± 0          | 0 ± 0          |
| task5                   | 0 ± 0             | 1 ± 1  | 0 ± 0   | 0 ± 0         | 0 ± 0          | 0 ± 0                  | 0 ± 0          | 0 ± 0        | <b>2 ± 1</b>  | 1 ± 1          | 1 ± 1          |
| <i>puzzle-3x3-play</i>  |                   |        |         |               |                |                        |                |              |               |                |                |
| task1                   | 5 ± 2             | 33 ± 6 | 97 ± 4  | 94 ± 3        | 90 ± 4         | 5 ± 0                  | 71 ± 3         | 78 ± 8       | 99 ± 0        | 73 ± 2         | <b>100 ± 0</b> |
| task2                   | 1 ± 1             | 4 ± 3  | 1 ± 1   | 1 ± 2         | 16 ± 5         | 0 ± 0                  | 2 ± 2          | 5 ± 2        | 98 ± 2        | 23 ± 5         | <b>100 ± 0</b> |
| task3                   | 1 ± 1             | 3 ± 2  | 3 ± 1   | 0 ± 0         | 10 ± 3         | 0 ± 0                  | 0 ± 0          | 4 ± 3        | <b>97 ± 1</b> | 83 ± 1         | 86 ± 2         |
| task4*                  | 1 ± 1             | 2 ± 1  | 2 ± 1   | 0 ± 0         | 16 ± 5         | 0 ± 0                  | 0 ± 0          | 5 ± 5        | 84 ± 24       | 42 ± 5         | <b>94 ± 4</b>  |
| task5                   | 1 ± 0             | 3 ± 2  | 5 ± 3   | 0 ± 0         | 16 ± 3         | 0 ± 0                  | 0 ± 0          | 6 ± 5        | 58 ± 39       | 43 ± 4         | <b>87 ± 8</b>  |
| <i>puzzle-4x4-play</i>  |                   |        |         |               |                |                        |                |              |               |                |                |
| task1                   | 1 ± 1             | 12 ± 2 | 26 ± 4  | 49 ± 9        | 34 ± 8         | 0 ± 0                  | 41 ± 2         | 37 ± 32      | 36 ± 3        | 38 ± 6         | <b>73 ± 8</b>  |
| task2                   | 0 ± 0             | 7 ± 4  | 12 ± 4  | 4 ± 4         | 16 ± 5         | 0 ± 0                  | 12 ± 4         | 10 ± 10      | 27 ± 5        | 23 ± 5         | <b>51 ± 2</b>  |
| task3                   | 0 ± 0             | 9 ± 3  | 15 ± 3  | 50 ± 14       | 18 ± 5         | 0 ± 0                  | 45 ± 7         | 33 ± 29      | 30 ± 4        | 40 ± 4         | <b>68 ± 5</b>  |
| task4*                  | 0 ± 0             | 5 ± 2  | 10 ± 3  | 21 ± 11       | 11 ± 3         | 0 ± 0                  | 23 ± 2         | 12 ± 10      | 28 ± 5        | 28 ± 4         | <b>49 ± 10</b> |
| task5                   | 0 ± 0             | 4 ± 1  | 7 ± 3   | 2 ± 2         | 7 ± 3          | 0 ± 0                  | 16 ± 6         | 10 ± 8       | 13 ± 2        | <b>23 ± 3</b>  | 20 ± 6         |
| <i>scene-play</i>       |                   |        |         |               |                |                        |                |              |               |                |                |
| task1                   | 19 ± 6            | 94 ± 3 | 95 ± 2  | 98 ± 3        | <b>100 ± 0</b> | 17 ± 3                 | <b>100 ± 0</b> | 99 ± 0       | 99 ± 0        | <b>100 ± 0</b> | <b>100 ± 0</b> |
| task2*                  | 1 ± 1             | 12 ± 3 | 50 ± 13 | 0 ± 0         | 76 ± 9         | 2 ± 1                  | 1 ± 0          | 85 ± 4       | 97 ± 1        | 57 ± 13        | <b>100 ± 0</b> |
| task3                   | 1 ± 1             | 32 ± 7 | 55 ± 16 | 54 ± 19       | 98 ± 1         | 0 ± 1                  | 94 ± 2         | 90 ± 3       | 94 ± 2        | 98 ± 2         | <b>100 ± 0</b> |
| task4                   | 2 ± 2             | 0 ± 1  | 3 ± 3   | 0 ± 0         | 5 ± 1          | 2 ± 1                  | 3 ± 1          | 0 ± 0        | 7 ± 17        | 12 ± 3         | <b>92 ± 8</b>  |
| task5                   | 0 ± 0             | 0 ± 0  | 0 ± 0   | 0 ± 0         | 0 ± 0          | 0 ± 0                  | 0 ± 0          | 0 ± 0        | 0 ± 0         | <b>2 ± 1</b>   | 0 ± 0          |

**Table 7:** Full offline evaluation on state-based OGBench tasks. Results cover 25 tasks and eight training seeds. An asterisk marks the domain-level tuning task. Bold denotes the highest displayed task mean; ties at the highest mean are all bold, while fully tied rows are left unbolded.

| Task                                    | Gaussian Policies |               | Flow Policies  |               | Distributional Critics |               |               |                |
|-----------------------------------------|-------------------|---------------|----------------|---------------|------------------------|---------------|---------------|----------------|
|                                         | IQL               | ReBRAC        | IFQL           | FQL           | IQN                    | Value Flows   | PCBF          | PA-BFM         |
| <i>visual-antmaze-medium-navigate</i>   |                   |               |                |               |                        |               |               |                |
| task1*                                  | 78 ± 9            | 54 ± 15       | <b>81 ± 3</b>  | 32 ± 3        | 62 ± 7                 | 77 ± 4        | 65 ± 3        | 78 ± 7         |
| task2                                   | 90 ± 3            | <b>96 ± 1</b> | 87 ± 1         | 60 ± 2        | 88 ± 2                 | 75 ± 5        | 73 ± 1        | 92 ± 2         |
| task3                                   | 80 ± 6            | 97 ± 1        | 92 ± 1         | 35 ± 8        | 64 ± 5                 | 81 ± 7        | 86 ± 4        | <b>98 ± 2</b>  |
| task4                                   | 89 ± 4            | 93 ± 2        | 84 ± 3         | 35 ± 2        | 71 ± 6                 | 71 ± 6        | 78 ± 3        | <b>96 ± 3</b>  |
| task5                                   | 84 ± 2            | <b>97 ± 1</b> | 89 ± 2         | 29 ± 7        | 84 ± 1                 | 70 ± 26       | 71 ± 1        | 93 ± 1         |
| <i>visual-antmaze-teleport-navigate</i> |                   |               |                |               |                        |               |               |                |
| task1*                                  | 5 ± 2             | 2 ± 0         | 7 ± 4          | 2 ± 1         | 2 ± 1                  | 10 ± 4        | 8 ± 2         | <b>21 ± 9</b>  |
| task2                                   | 10 ± 2            | 10 ± 3        | 13 ± 3         | 6 ± 1         | 7 ± 3                  | 17 ± 5        | 19 ± 4        | <b>33 ± 8</b>  |
| task3                                   | 7 ± 7             | 4 ± 1         | 8 ± 9          | 9 ± 4         | 6 ± 4                  | 16 ± 3        | <b>18 ± 4</b> | 16 ± 2         |
| task4                                   | 4 ± 6             | 4 ± 0         | 18 ± 2         | 9 ± 1         | 4 ± 2                  | 16 ± 5        | 18 ± 5        | <b>32 ± 3</b>  |
| task5                                   | 2 ± 1             | 2 ± 1         | 4 ± 2          | 1 ± 1         | 2 ± 1                  | 8 ± 2         | 8 ± 3         | <b>10 ± 2</b>  |
| <i>visual-cube-double-play</i>          |                   |               |                |               |                        |               |               |                |
| task1*                                  | 34 ± 23           | 4 ± 4         | 8 ± 6          | 23 ± 4        | 4 ± 1                  | <b>35 ± 2</b> | 10 ± 3        | 30 ± 10        |
| task2                                   | 3 ± 1             | 0 ± 0         | 0 ± 0          | 0 ± 0         | 0 ± 0                  | <b>4 ± 2</b>  | 0 ± 0         | 0 ± 0          |
| task3                                   | 7 ± 4             | 2 ± 2         | 1 ± 1          | 4 ± 2         | 0 ± 0                  | <b>11 ± 2</b> | 0 ± 0         | 10 ± 4         |
| task4                                   | 2 ± 1             | 0 ± 0         | 0 ± 0          | 0 ± 0         | 0 ± 0                  | 2 ± 1         | 0 ± 0         | <b>10 ± 4</b>  |
| task5                                   | 11 ± 2            | 0 ± 0         | 2 ± 1          | 4 ± 1         | 1 ± 1                  | 13 ± 3        | 3 ± 1         | <b>33 ± 9</b>  |
| <i>visual-puzzle-3x3-play</i>           |                   |               |                |               |                        |               |               |                |
| task1*                                  | 7 ± 15            | 88 ± 4        | <b>100 ± 0</b> | 94 ± 1        | 84 ± 1                 | 93 ± 5        | 81 ± 1        | 90 ± 2         |
| task2                                   | 0 ± 0             | <b>12 ± 1</b> | 0 ± 0          | 0 ± 0         | 6 ± 2                  | <b>12 ± 1</b> | 0 ± 0         | 0 ± 0          |
| task3                                   | 0 ± 0             | 1 ± 1         | 2 ± 1          | 0 ± 0         | 1 ± 0                  | 3 ± 1         | 2 ± 1         | <b>5 ± 1</b>   |
| task4                                   | 1 ± 1             | 0 ± 1         | 1 ± 0          | 5 ± 4         | 3 ± 0                  | <b>6 ± 2</b>  | 5 ± 3         | 4 ± 1          |
| task5                                   | 0 ± 0             | 0 ± 0         | 0 ± 0          | 1 ± 0         | 1 ± 0                  | <b>2 ± 0</b>  | 1 ± 0         | <b>2 ± 1</b>   |
| <i>visual-scene-play</i>                |                   |               |                |               |                        |               |               |                |
| task1*                                  | 97 ± 2            | 98 ± 4        | 86 ± 10        | 98 ± 3        | 95 ± 2                 | 99 ± 0        | 40 ± 4        | <b>100 ± 1</b> |
| task2                                   | 21 ± 16           | 30 ± 15       | 0 ± 0          | <b>86 ± 8</b> | 79 ± 15                | 40 ± 27       | 62 ± 8        | 42 ± 14        |
| task3                                   | 12 ± 9            | 10 ± 7        | 19 ± 2         | 22 ± 6        | 31 ± 14                | 66 ± 1        | 35 ± 1        | <b>68 ± 12</b> |
| task4                                   | 1 ± 0             | 0 ± 0         | 0 ± 0          | 1 ± 1         | 0 ± 0                  | 10 ± 6        | 12 ± 4        | <b>38 ± 11</b> |
| task5                                   | 0 ± 0             | 0 ± 0         | 0 ± 0          | 0 ± 0         | 0 ± 0                  | 0 ± 0         | 0 ± 0         | 0 ± 0          |

**Table 8:** Full offline evaluation on 25 image-based OGBench tasks over four seeds. An asterisk marks the tuning task; bold denotes the highest displayed task mean, with fully tied rows left unbolded.

| Method                                   | Antmaze     | Cube-D      | Cube-T       | Scene        |
|------------------------------------------|-------------|-------------|--------------|--------------|
| <b>Offline checkpoint at 1000k steps</b> |             |             |              |              |
| IQN                                      | 55±5        | 29±4        | 25±5         | 0±0          |
| IFQL                                     | 11±3        | 12±3        | 2±1          | 0±0          |
| FQL                                      | <b>87±7</b> | 40±11       | 4±1          | <b>82±11</b> |
| RLPD                                     | 0±0         | 0±0         | 0±0          | 0±0          |
| IQL                                      | 41±5        | 0±0         | 3±2          | 15±4         |
| Value Flows                              | 72±4        | 65±7        | <b>29±8</b>  | <b>83±4</b>  |
| PA-BFM                                   | <b>89±4</b> | <b>78±4</b> | 24±2         | <b>79±3</b>  |
| <b>Online final at 2000k steps</b>       |             |             |              |              |
| IQN                                      | 91±1        | 42±7        | 14±6         | 0±0          |
| IFQL                                     | 3±2         | 41±2        | 7±5          | 60±14        |
| FQL                                      | <b>99±1</b> | <b>92±3</b> | <b>83±12</b> | <b>100±1</b> |
| RLPD                                     | 3±2         | 0±0         | 0±0          | <b>100±0</b> |
| IQL                                      | 45±5        | 0±0         | 0±0          | 10±3         |
| Value Flows                              | <b>96±1</b> | 79±6        | 70±7         | <b>100±0</b> |
| PA-BFM                                   | <b>99±1</b> | <b>95±2</b> | <b>85±4</b>  | <b>100±0</b> |

**Table 9:** Offline-to-online evaluation on four OGBench scenarios. The two panels report success at online entry and after one million online environment steps. Entries are means  $\pm$  standard deviations over eight seeds; bold denotes means within 95% of the highest scenario mean.

analysis in Appendix B.5. Table 10 reports the complete fixed-strength sweep. Stronger fixed tilting reduces positive overestimation but eventually degrades policy performance. The adaptive variant reduces normalized positive overestimation from 0.380 to 0.165 while increasing final success from 0.490 to 0.515.

| Variant            | Positive overestimation ↓           | Final success ↑                     |
|--------------------|-------------------------------------|-------------------------------------|
| No distortion      | 0.380 $\pm$ 0.033                   | 0.490 $\pm$ 0.100                   |
| Fixed $\eta = 0.1$ | 0.300 $\pm$ 0.027                   | 0.505 $\pm$ 0.090                   |
| Fixed $\eta = 0.3$ | 0.220 $\pm$ 0.024                   | 0.455 $\pm$ 0.110                   |
| Fixed $\eta = 0.6$ | <b>0.105 <math>\pm</math> 0.019</b> | 0.365 $\pm$ 0.130                   |
| Adaptive           | 0.165 $\pm$ 0.023                   | <b>0.515 <math>\pm</math> 0.090</b> |
| Value Flows        | 0.335 $\pm$ 0.041                   | 0.280 $\pm$ 0.050                   |

**Table 10:** Conservatism–performance trade-off on `puzzle-4x4-play-task4`. The undistorted PA-BFM and Value Flows scores correspond to the task-level results in the main benchmark table. Lower positive overestimation and higher final success are better.

## D.5 Runtime Analysis

Component and pool-size ablations are shown in Figure 3 of the main paper. The corresponding final-score differences are reported in the main text; here we focus on computational cost.

**Computational cost.** Table 11 reports wall-clock hours for  $10^6$ -update budgets on one representative state-based configuration and one representative visual configuration. These are configuration-level values rather than averages over all benchmark domains. All methods use one NVIDIA GeForce RTX 4090 with 24 GB of memory, batch size 256, and the same implementation, compilation, and

data-loading pipeline. The same end-to-end timing boundary is applied to every method within a column; endpoint candidates within an update are evaluated in parallel and reuse the successor-state representation.

| Method                        | Target work       | State | Visual |
|-------------------------------|-------------------|-------|--------|
| BCFM ( $K = 1$ )              | 2 target solves   | 3.6   | 10.5   |
| Uniform ( $K = 8$ )           | 16 target solves  | 3.8   | 11.0   |
| PA-BCFM ( $K = 4$ )           | 8 solves + wt.    | 3.8   | 11.0   |
| PA-BCFM ( $K = 8$ )           | 16 solves + wt.   | 4.0   | 11.8   |
| PA-BCFM ( $K = 16$ )          | 32 solves + wt.   | 4.5   | 13.2   |
| Value Flows                   | path + confidence | 4.4   | 12.5   |
| PCBF ( $\lambda_{cv} = 0.3$ ) | endpoint + CV     | 4.0   | 11.5   |

**Table 11:** Wall-clock training cost. State and visual columns report configuration-level hours for  $10^6$ -update budgets on one NVIDIA GeForce RTX 4090 (24 GB) under the two representative timing settings. Target work counts both target critics, so  $K$  ensemble-mean endpoints require  $2K$  target-flow solves.

The default  $K = 8$  configuration remains below the measured Value Flows runtime in both settings;  $K = 16$  adds cost with little gain in Figure 3, supporting  $K = 8$  as the default.

## E Experimental Details

All methods are implemented and evaluated in a common JAX codebase adapted from FQL and Value Flows (Park et al., 2025b; Dong et al., 2026). Methods share OGBench wrappers, dataset revisions, preprocessing, budgets, and evaluation code. BCFM, Value Flows, PCBF, and PA-BFM additionally share the actor, behavior proposal, critic backbone, target update, ODE solver, schedule, and mean-based action extraction. They differ in critic supervision: endpoint matching, weighted consistency, a path-coupled control variate, and endpoint-pool posterior averaging, respectively. Isolation claims apply only to this group.

### E.1 Environments

**Controlled environments.** The one-state Bernoulli-return process is a development process used for configuration selection. Solitaire Dice and the discrete Monte Carlo chain are the controlled evaluation processes. The complete process definitions are given in Table 3. All three processes admit exact return laws, so the reference return distributions and posterior Bellman velocities are computed from the exact endpoint laws; no Monte Carlo reference pool is used for the controlled results.

For Discrete-MC, let  $n = 20$ ,  $\mathcal{X} = \{0, \dots, n - 1\}$ , and define the unnormalized state weight

$$\omega_i = \exp\left[\frac{n-1}{4\pi} \cos\left(\frac{4\pi(i-1)}{n-1}\right)\right]. \quad (121)$$

For each nonterminal state  $i \in \{1, \dots, n - 2\}$ , the transition kernel is

$$\begin{aligned} P(i, i \pm 1) &= \frac{1}{2} \frac{\omega_{i \pm 1}}{\omega_i + \omega_{i \pm 1}}, \\ P(i, i) &= 1 - P(i, i - 1) - P(i, i + 1), \end{aligned} \quad (122)$$

with  $P(0, 0) = P(n - 1, n - 1) = 1$ . Episodes initialize from  $X_0 \sim \text{Unif}\{1, \dots, n - 2\}$ . A transition receives reward  $R_h = \mathbf{1}\{X_{h+1} \notin \{0, n - 1\}\}$ , and  $\gamma = 0.95$ .

**Offline OGBench.** We evaluate the 50 single-task state and image environments listed in Tables 7–8 using the standard OGBench protocol (Dong et al., 2026; Park et al., 2025a). State-based tasks use native observations; image-based tasks use three stacked RGB frames, a small IMPALA encoder, and random-crop augmentation.

**Offline-to-online OGBench.** We additionally evaluate offline-to-online fine-tuning on `antmaze-large`, `cube-double`, `cube-triple`, and `scene`. Each method is first trained for one million offline optimizer updates and is then fine-tuned for one million online environment steps while newly collected transitions are appended to the replay buffer. Each method retains its own critic objective throughout both phases.

## E.2 Methods for Comparison

The offline tables compare Gaussian-policy scalar critics (BC, IQL, and ReBRAC), flow-policy scalar critics (IFQL and FQL), distributional critics (C51, IQN, and CODAC), and flow-based distributional critics (Value Flows, PCBF, and PA-BFM) (Kostrikov et al., 2022; Tarasov et al., 2023; Park et al., 2025b; Bellemare et al., 2017; Dabney et al., 2018a; Ma et al., 2021; Dong et al., 2026; Xu et al., 2026). PCBF also appears in the controlled and runtime comparisons but not in offline-to-online evaluation. The latter additionally includes RLPD under the common interaction budget and checkpoint schedule (Ball et al., 2023). PA-BFM, Value Flows, and PCBF share the policy-side implementation in the matched comparison and differ in critic supervision.

## E.3 Evaluation Protocol

**Offline evaluation.** State-based and image-based tasks use eight and four training seeds, respectively. Each evaluation checkpoint contains 50 episodes, and the main tables report the final checkpoint. For each domain, the reported mean is the arithmetic mean of the five task means, and the displayed uncertainty is the arithmetic mean of the corresponding five task-level standard deviations. The task marked with an asterisk is used for domain-level hyperparameter selection under a common search budget for the matched methods. The selected configuration is fixed across the remaining tasks in that domain, while the tuning task remains in the five-task aggregate for comparability with prior work. Task-level win/tie/loss counts compare final means and are descriptive rather than significance tests.

**Offline-to-online evaluation.** Online interaction begins from the one-million-optimizer-update offline checkpoint. Methods are evaluated every 100k environment steps through 2000k total steps. Each method uses eight independent training seeds. At each checkpoint, the learning curves show the mean success rate over training seeds, and shaded bands show one standard deviation across training seeds. Training seeds and evaluation environment seeds are disjoint, and evaluation seeds are shared across methods within each scenario.

**Controlled evaluation.** The controlled posterior-target and gradient-covariance protocols are specified with their corresponding metrics in Appendix D.1. Paired estimator comparisons reuse the same candidate endpoint pools whenever the estimators share an endpoint budget. Controlled target-MSE and gradient-covariance intervals use a process-stratified query-cluster bootstrap, with all replicates of a selected query retained together. Learned-fidelity intervals use training seeds as the resampling unit, and paired method differences use matched seed indices. All reported intervals are 95% confidence intervals.

| Hyperparameter                                        | Value                             |
|-------------------------------------------------------|-----------------------------------|
| Optimizer                                             | Adam (Kingma and Ba, 2015)        |
| Batch size                                            | 256                               |
| Critic learning rate                                  | $3 \times 10^{-4}$                |
| Critic hidden-layer sizes                             | (512, 512, 512, 512)              |
| Activation function                                   | GELU                              |
| Actor layer normalization                             | Yes                               |
| Critic/value layer normalization                      | Yes                               |
| Return-flow Euler steps                               | 10                                |
| Return samples per candidate action $M_{\text{act}}$  | 1                                 |
| Candidate actions $N$                                 | 16                                |
| Target-network update coefficient $\tau_{\text{tgt}}$ | $5 \times 10^{-3}$                |
| Number of critics                                     | 2                                 |
| Image encoder                                         | Small IMPALA (Park et al., 2025b) |
| Image augmentation                                    | Random cropping                   |
| Image frame stack                                     | 3                                 |
| Numeric precision                                     | float32                           |

**Table 12:** Common hyperparameters for PA-BFM and matched flow-critic baselines. Entries apply where the corresponding component is present.

## E.4 Architectures and Hyperparameters

The network architectures and optimization settings of the matched flow-critic methods follow Value Flows (Dong et al., 2026). Table 12 summarizes the settings shared by PA-BFM and the corresponding flow-critic baselines. Entries involving flow integration or candidate-action selection apply only to methods containing those components.

**PA-BFM-specific hyperparameters.** The default PA-BCFM critic uses  $K = 8$  Bellman endpoints and no pessimistic distortion, i.e.,  $g \equiv 1$ . Bridge time is sampled as  $t \sim \text{Unif}(0, 1 - 10^{-5})$ , and posterior logits are normalized by log-sum-exp. All downstream experiments use this setting;  $K \in \{1, 2, 4, 8, 16\}$  is varied only in the controlled estimator and pool-sensitivity studies.

**Baseline hyperparameter selection.** IQL selects its advantage-weighting inverse temperature from  $\{0.3, 1, 3, 10\}$ . ReBRAC selects its actor and critic regularization coefficients from  $\{0.003, 0.01, 0.03, 0.1, 0.3, 1\}$  and  $\{0, 0.001, 0.01, 0.1\}$ , respectively. IFQL selects the number of proposed actions from  $\{32, 64, 128\}$ , and FQL uses its original regularization grid. C51 uses 51 or 101 atoms. IQN and CODAC select the quantile-loss parameter from  $\{0.7, 0.8, 0.9, 0.95\}$ , with CODAC penalty 0.1. PCBF uses  $\lambda_{\text{cv}} = 0.3$ , selected on the Bernoulli development process and frozen for Solitaire Dice and Discrete-MC. Each selected setting is fixed for the remaining tasks in its group.

## E.5 Runtime and Reproducibility

All experiments use JAX on one NVIDIA GeForce RTX 4090 (24 GB), with batch size 256 and common implementation, compilation, data-loading, and evaluation boundaries. Table 11 reports end-to-end wall-clock hours for one million updates on representative state and visual configurations; these are configuration-level measurements, not domain averages. Performance comparisons use matched optimizer-update counts.

We record the software and hardware versions, OGBench and dataset revisions, random seeds, training commands, configuration overrides, checkpoint-selection rules, and evaluation commands. Tables and figures use the same plotting and aggregation pipeline across methods.

| Domain/scenario         | Evaluation | $\gamma$ |
|-------------------------|------------|----------|
| cube-double             | Off.+O2O   | .995     |
| cube-triple             | Off.+O2O   | .995     |
| puzzle-3x3              | Offline    | .99      |
| puzzle-4x4              | Offline    | .99      |
| scene                   | Off.+O2O   | .99      |
| visual-antmaze-medium   | Offline    | .99      |
| visual-antmaze-teleport | Offline    | .99      |
| visual-cube-double      | Offline    | .995     |
| visual-puzzle-3x3       | Offline    | .99      |
| visual-scene            | Offline    | .99      |
| antmaze-large           | O2O only   | .99      |

**Table 13:** Domain/scenario settings. PA-BFM uses an undistorted  $K = 8$  critic, and all main experiments use mean-based candidate scoring.