FLOW-TIME BELLMAN MATCHING FOR OFFLINE DISTRIBUTIONAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributional reinforcement learning models the full return distribution rather than only its expectation, providing a richer representation of future returns for value learning. Flow matching provides a flexible framework for modeling return distributions, but existing approaches typically impose Bellman consistency only on the terminal return distribution, leaving the temporal structure of the transport process unused. We introduce *Flow-Time Bellman Matching* (FBM), which extends Bellman regression from the terminal return distribution to the entire return flow by constructing Bellman targets from the successor flow at each corresponding flow time. We show that the proposed flow-time Bellman operator is consistent with the standard distributional Bellman operator at the endpoint. To enable efficient evaluation, we further distill the return flow into a one-step quantile critic using an exact correspondence between Gaussian source noise and return quantiles. Experiments on OGBench and D4RL show that FBM achieves the best or near-best performance on 25 of 37 state-based tasks, while controlled analyses demonstrate the contribution of flow-time Bellman supervision and quantile distillation.

1 INTRODUCTION

Accurate critic learning is a central challenge in offline reinforcement learning (RL), where value functions must be learned from a fixed dataset (Lange et al., 2012; Levine et al., 2020; Kumar et al., 2019).A common approach in offline RL is the actor–critic framework, where the critic estimates the expected return to guide policy improvement (Wu et al., 2019; Fujimoto & Gu, 2021; Nair et al., 2021; Tarasov et al., 2023). However, such expectation-based value estimation compresses the full return distribution into a single scalar, discarding information about return variability, multimodality, and tail behavior that can provide richer signals for value estimation (Morimura et al., 2010; Bellemare et al., 2017; Dabney et al., 2018b; Bellemare et al., 2023). This limitation is particularly relevant in offline RL, where distribution shifts and limited dataset coverage make reliable value estimation challenging (Kumar et al., 2020; Fu et al., 2021).

Distributional reinforcement learning provides a natural extension of value estimation by modeling the full return distribution, providing a richer representation for value learning and policy improvement (Morimura et al., 2010; Bellemare et al., 2023; Mavrin et al., 2019; Bellemare et al., 2017; Dabney et al., 2018b). Existing distributional methods often rely on structured parameterizations of return distributions, such as categorical projections onto fixed supports (Bellemare et al., 2017; Rowland et al., 2018) or quantile-based representations (Dabney et al., 2018b;a). While effective, these approaches impose predefined structures on the representation of the return distribution. Flow matching provides a more flexible alternative by learning a continuous transport from a tractable source distribution to the target return distribution, enabling expressive modeling of complex and multimodal distributions in continuous spaces (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023; Lipman et al., 2024). This flexibility has motivated its recent adoption for return distribution modeling (Farebrother et al., 2025; Dong et al., 2025).

However, existing flow-based return critics do not fully exploit the structure of the learned return flow during Bellman learning. While flow matching defines a continuous transport trajectory with intermediate return distributions, existing flow-based return critics typically impose Bellman supervision through sampled terminal successor returns (Agrawalla et al., 2026; Dong et al., 2025). This endpoint construction increases the variance of the flow-matching regression targets, making the value

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

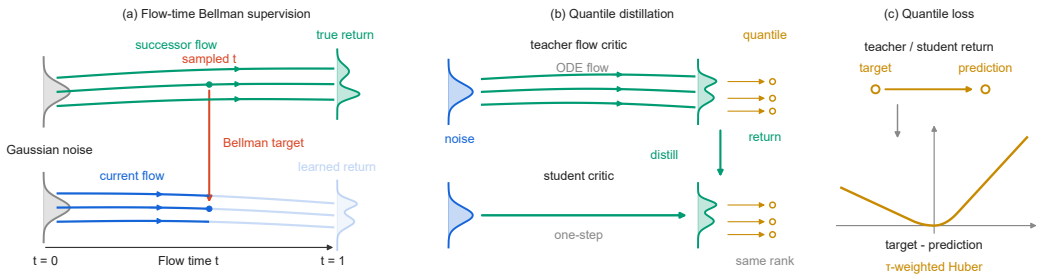

Figure 1: **Overview of FBM.** Left: flow-time Bellman supervision constructs the training target for the current return flow from the successor flow at the sampled flow time. Middle: the order-preserving return flow induces a noise–quantile correspondence, enabling one-step quantile distillation. Right: the distilled critic matches the corresponding teacher return using a quantile-weighted Huber loss.

estimation more difficult and potentially degrading the accuracy of the learned return flow. Since the successor return flow is already defined throughout the transport trajectory, a natural question arises: can Bellman supervision be extended from the terminal distribution to the entire flow?

In this paper, we introduce *Flow-Time Bellman Matching* (FBM), which extends Bellman regression from the terminal distribution to the entire return flow. Our key idea is to define the distributional Bellman backup directly along the probability path induced by the return flow. Specifically, at each sampled flow time t, FBM evaluates the successor flow at the corresponding time and uses its vector field to construct the Bellman regression target for the current return flow. We formalize this construction through a flow-time Bellman operator. Theoretically, we show that the flow-time Bellman operator is a γ-contraction in the maximal p-Wasserstein metric at every flow time, which recovers the standard distributional Bellman operator at $t = 1$. As a result, the flow-time extension retains the fixed-point convergence guarantee of distributional Bellman evaluation. Moreover, by replacing sampled terminal successor returns with the successor vector field, FBM reduces the variance of Bellman velocity targets, facilitating more accurate return-flow learning.

We further show that return flows preserve the ordering of source samples throughout the transport process, inducing an exact correspondence between Gaussian source noise and return quantiles. This correspondence enables us to distill the iterative flow critic into an explicit one-step quantile critic by matching the same quantile across the flow and the distilled critic. However, because the distilled critic selects the actions used in subsequent Bellman backups, any lag under pure distillation can propagate into return-flow training and degrade policy improvement and critic learning. We supplement the distillation objective with a direct quantile Bellman objective, providing an additional signal for faster critic adaptation. This design allows the one-step critic to track the return flow more closely while retaining efficient value evaluation without repeated numerical integration. As illustrated in Figure 1, FBM extends Bellman supervision throughout the entire return flow and further constructs an efficient one-step critic for policy evaluation.

We evaluate FBM on OGBench (Park et al., 2024) and D4RL Adroit (Fu et al., 2021) tasks, where FBM achieves the highest success rate on 25/37 tasks. Controlled analyses further isolate the effect of flow-time Bellman supervision and study the sensitivity of quantile distillation and candidate-based policy extraction.

2 Related Work

Offline reinforcement learning. Offline reinforcement learning learns policies from fixed datasets without additional environment interaction, where distribution shift between the dataset and the learned policy can cause inaccurate value estimation and extrapolation errors on out-of-distribution actions (Levine et al., 2020; Kumar et al., 2019). Early approaches mitigate this challenge through conservative value estimation (Kumar et al., 2020), behavior regularization (Fujimoto et al., 2019; Wu et al., 2019), and implicit value learning (Kostrikov et al., 2021). Recent methods have explored improved policy optimization strategies for offline RL, including improved actor–critic optimization

in ReBRAC (Tarasov et al., 2023), flow-based policy learning in FQL (Park et al., 2025), and behavior-regularized diffusion policies (Gao et al., 2025).

Distributional reinforcement learning. Distributional reinforcement learning models the full return distribution rather than only its expectation, providing richer information about return variability and tail behavior for value learning (Bellemare et al., 2023). Existing methods represent return distributions through structured parameterizations, including categorical distributions, quantile functions, and implicit quantile representations (Bellemare et al., 2017; Dabney et al., 2018b;a). These distributional representations have been applied to risk-sensitive decision making problems (Ma et al., 2025; Lim & Malik, 2022; Ma et al., 2021), and recent theoretical results further suggest benefits beyond risk-sensitive objectives (Wang et al., 2023a; 2024). In contrast to atom- or quantile-parameterized critics, FBM represents the return distribution with a one-dimensional continuous flow and exploits its induced quantile structure to construct an explicit policy-facing critic.

Generative models in offline reinforcement learning. Expressive generative models have increasingly been used in offline reinforcement learning to capture complex and multimodal behavior distributions. Diffusion-based methods combine generative behavior modeling with value-guided optimization or sampling-based policy extraction (Chen et al., 2022; Hansen-Estruch et al., 2023; Wang et al., 2023b; Kang et al., 2023; Chen et al., 2024; Ding et al., 2024), while flow-based methods provide an ODE-based alternative for behavior modeling and policy optimization (Park et al., 2025; Zhang et al., 2025; Tiofack et al., 2026). Different from existing approaches that primarily employ generative models for policy modeling, FBM applies generative modeling to the critic to learn the return distribution.

Generative models have recently been explored for temporal-difference learning by modeling state-conditioned predictions and incorporating Bellman-style bootstrapping into generative processes (Janner et al., 2021; Farebrother et al., 2025). Flow-based generative critics have been investigated for both scalar value estimation and distributional return modeling in recent works (Agrawalla et al., 2026; Dong et al., 2025). FBM follows this generative critic direction but differs by introducing Bellman supervision throughout the return-flow path and exploiting the induced quantile structure for offline RL.

3 PRELIMINARIES

Problem Setting. We consider a discounted Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \rho_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $p(s' \mid s, a)$ is the transition kernel, $r(s, a)$ is the reward function, ρ_0 is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor. A policy $\pi(a \mid s)$ induces trajectories through $S_0 \sim \rho_0$, $A_t \sim \pi(\cdot \mid S_t)$, and $S_{t+1} \sim p(\cdot \mid S_t, A_t)$. The objective is to maximize the expected discounted return

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \right]. \tag{1}$$

In offline reinforcement learning, the learner is given a fixed dataset $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$ collected by an unknown behavior policy π_β and must learn a policy π without additional environment interaction.

Distributional Reinforcement Learning. Distributional reinforcement learning models the entire return distributions. For a fixed policy π, we define the conditional return random variable starting from (s, a) as $Z^\pi(s, a) = \sum_{h=0}^{\infty} \gamma^t r(S_t, A_t)$ with $S_0 = s,\ A_0 = a$, Its expectation recovers the usual action-value function:$Q^\pi(s, a) = \mathbb{E}[Z^\pi(s, a)]$. The distributional Bellman operator is

$$(\mathcal{T}^\pi Z)(s, a) \overset{D}{=} r(s, a) + \gamma Z(S', A'), \tag{2}$$

where $S' \sim p(\cdot \mid s, a)$, $A' \sim \pi(\cdot \mid S')$, and $\overset{D}{=}$ denotes equality in distribution. The true return distribution Z^π is the fixed point of this operator.

Flow Matching for Return Modeling. Flow matching learns a continuous transport from a source distribution to a target distribution through a time-dependent vector field (Lipman et al., 2023; 2024). In our setting, the source is a standard Gaussian and the target is the conditional return distribution.

For each state–action pair (s, a), we parameterize a one-dimensional vector field $v_\theta(z, t, s, a)$ whose flow map $\phi^{s,a}_{\theta,t}$ satisfies

$$\frac{d}{dt}\phi^{s,a}_{\theta,t}(\epsilon) = v_\theta\Big(\phi^{s,a}_{\theta,t}(\epsilon), t, s, a\Big), \qquad \phi^{s,a}_{\theta,0}(\epsilon) = \epsilon, \quad \epsilon \sim \mathcal{N}(0,1), \tag{3}$$

Here $t \in [0,1]$ denotes the flow time of the generative transport; the underlying MDP and Bellman recursion remain discrete-time. Given a target return sample z_1, conditional flow matching constructs the linear interpolation

$$z_t = (1-t)\epsilon + tz_1, \qquad t \sim \mathrm{Unif}([0,1]), \tag{4}$$

whose conditional velocity is $z_1 - \epsilon$. The vector field is therefore trained with

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}\left[\|v_\theta(z_t, t, s, a) - (z_1 - \epsilon)\|_2^2\right]. \tag{5}$$

After training, integrating Eq. (3) from $t = 0$ to $t = 1$ transforms Gaussian noise into samples from the learned return distribution. In RL, however, samples from the target return distribution are not directly available and must instead be obtained through Bellman bootstrapping.

4 Flow-Time Bellman Matching

We now introduce *Flow-Time Bellman Matching* (FBM), a flow-based distributional RL method for offline control. FBM performs Bellman regression throughout the return flow by constructing targets from the successor flow at the corresponding flow time. We further exploit the order-preserving property of one-dimensional flows to distill the iterative flow critic into an efficient one-step quantile critic.

4.1 Flow-time Bellman Supervision

A direct way to train a flow-based return critic is to sample a terminal successor return from the target flow, construct the Bellman target, and use it for conditional flow matching. For a transition (s, a, r, s') and a next action $a' \sim \pi(\cdot \mid s')$, this procedure samples $Z'_1 \sim Z(s', a')$ and forms the terminal target $Y = r(s, a) + \gamma Z'_1$. The critic is then trained by interpolating between Gaussian source noise and Y.

Although conditional flow matching optimizes the vector field at intermediate flow times, the corresponding training states are generated by interpolating toward the terminal Bellman target Y. Our key idea is instead to impose Bellman supervision directly at each flow time, so that the current return flow is trained using a Bellman target constructed from the successor return flow at the same time t, rather than only from a terminal successor return.

We begin with the linear interpolation used in conditional flow matching (Lipman et al., 2023). For a successor return variable Z'_1 and source noise $\xi \sim \mathcal{N}(0,1)$, define $Z_t = tZ'_1 + (1-t)\xi, \quad \xi \sim \mathcal{N}(0,1)$, which interpolates between the Gaussian source at $t = 0$ and the return distribution at $t = 1$. Applying the same interpolation to the Bellman target $r(s, a) + \gamma Z'_1$ gives

$$Z_t = t\big(r(s,a) + \gamma Z'_1\big) + (1-t)\xi. \tag{6}$$

Flow-time Bellman operator. Our goal is to express the interpolated Bellman target in Eq. (6) using the successor flow at the same flow time t. To preserve the Gaussian source while exposing the successor interpolation, let $\xi', \eta \sim \mathcal{N}(0,1)$ be independent and write $\xi \overset{D}{=} \gamma\xi' + \sqrt{1-\gamma^2}\eta$. The right-hand side follows $\mathcal{N}(0, \gamma^2 + (1-\gamma^2)) = \mathcal{N}(0,1)$ and therefore has the same distribution as the original source noise ξ. Substituting this representation into Eq. (6) and regrouping terms gives

$$\begin{aligned} t\big(r(s,a) + \gamma Z'_1\big) + (1-t)\xi &\overset{D}{=} tr(s,a) + \gamma t Z'_1 + \gamma(1-t)\xi' + (1-t)\sqrt{1-\gamma^2}\eta \\ &= tr(s,a) + \gamma\big(tZ'_1 + (1-t)\xi'\big) + (1-t)\sqrt{1-\gamma^2}\eta. \end{aligned} \tag{7}$$

The term $tZ'_1 + (1-t)\xi'$ is precisely the linear interpolation between source noise ξ' and the successor return Z'_1, and therefore corresponds to the successor return flow at time t. This yields the flow-time Bellman operator, which constructs the distributional Bellman target directly at flow time t

$$\big(\mathcal{B}^\pi_t Z_t\big)(s,a) \overset{D}{=} tr(s,a) + \gamma Z_t(s', a') + (1-t)\sqrt{1-\gamma^2}\,\eta, \qquad \eta \sim \mathcal{N}(0,1). \tag{8}$$

Flow-matching losses. To obtain a tractable training objective, we express the Bellman update in Eq. (8) as a flow-matching regression objective. Let z'_t denote the successor return produced by the target critic at flow time t, i.e., $z'_t = \phi^{s',a'}_{\bar{\theta},t}(\xi')$, where $\bar{\theta}$ denotes the target critic parameters. The corresponding Bellman target at flow time t is

$$x_t = tr(s,a) + \gamma z'_t + (1-t)\sqrt{1-\gamma^2}\,\eta. \tag{9}$$

Since z'_t follows the flow ODE, $\frac{dz'_t}{dt} = v_{\bar{\theta}}(z'_t, t, s', a')$, differentiating x_t with respect to flow time gives the corresponding velocity target

$$u_t = r(s,a) + \gamma v_{\bar{\theta}}(z'_t, t, s', a') - \sqrt{1-\gamma^2}\,\eta. \tag{10}$$

This construction yields a velocity target u_t for the current return flow. We then train the return vector field to match u_t at x_t:

$$\mathcal{L}_{\mathrm{FBM}}(\theta) = \mathbb{E}\left[\left\|v_\theta(x_t, t, s, a) - \mathrm{sg}(u_t)\right\|_2^2\right], \tag{11}$$

where the expectation is over $(s,a,r,s') \sim \mathcal{D}$, $a' \sim \pi(\cdot \mid s')$, $t \sim \mathrm{Unif}[0,1]$, and $\xi', \eta \sim \mathcal{N}(0,1)$.

Theoretical properties. We next establish that the proposed flow-time Bellman construction preserves the correct return distribution and that the practical regression objective recovers the corresponding vector field. The proofs of all propositions are provided in Appendix B.

Proposition 1. *For every $t \in [0,1]$, $\mathcal{B}^\pi_t$ is a γ-contraction under the maximal p-Wasserstein metric. Its unique fixed point satisfies*

$$Z^\pi_t(s,a) \overset{D}{=} tZ^\pi(s,a) + (1-t)\xi, \qquad \xi \sim \mathcal{N}(0,1).$$

Moreover, $\mathcal{B}^\pi_1$ recovers the standard distributional Bellman operator, while $\mathcal{B}^\pi_0$ leaves the Gaussian source invariant.

Proposition 1 shows that extending Bellman supervision throughout flow time preserves the solution of standard distributional Bellman learning. The contraction guarantees a unique fixed point at every t, and this fixed point follows exactly the interpolation from the Gaussian source to the policy return distribution. In particular, the two boundary cases recover the prescribed source distribution at $t = 0$ and the standard distributional Bellman fixed point at $t = 1$.

Proposition 2 (Consistency of Bellman flow matching)**.** *Let $p_{\bar{\theta},t}(\cdot \mid s,a)$ denote the probability path generated by the target flow. Assume that x_t is almost surely absolutely continuous with $\dot{x}_t = u_t$ and that $\mathbb{E}\left[\int_0^1 |u_t|^2\,dt\right] < \infty$. Then the population minimizer of Eq.* (11) *over square-integrable vector fields satisfies*

$$v^\star(x,t,s,a) = \mathbb{E}[u_t \mid x_t = x, s, a],$$

and $v^\star$ generates the Bellman-updated probability path

$$q(x_t \mid s,a) = \left(\mathcal{B}^\pi_t p_{\bar{\theta},t}\right)(\cdot \mid s,a).$$

Proposition 2 connects the practical flow-matching objective to the distributional operator in Proposition 1. Although u_t is a stochastic velocity target, regression recovers its conditional mean at the population optimum, and this vector field transports the Bellman-updated return distribution throughout flow time.

Proposition 3. *Let $Z'_t = tZ'_1 + (1-t)\xi', \quad D' = Z'_1 - \xi'$, and assume that D' has finite second moment. Define $v^\star_{\mathrm{suc}}(Z'_t, t, s', a') = \mathbb{E}[D' \mid Z'_t, s', a']$, and let*

$$u^{\mathrm{term}}_t = r + \gamma D' - \sqrt{1-\gamma^2}\eta, \qquad u^{\mathrm{flow}}_t = r + \gamma v^\star_{\mathrm{suc}}(Z'_t, t, s', a') - \sqrt{1-\gamma^2}\eta,$$

where $\eta \sim \mathcal{N}(0,1)$ is independent of the successor coupling. Then

$$\mathbb{E}\left[u^{\mathrm{flow}}_t \mid s,a,r,s',a'\right] = \mathbb{E}\left[u^{\mathrm{term}}_t \mid s,a,r,s',a'\right],$$

and

$$\mathrm{Var}\left(u^{\mathrm{flow}}_t \mid s,a,r,s',a'\right) \le \mathrm{Var}\left(u^{\mathrm{term}}_t \mid s,a,r,s',a'\right).$$

Proposition 3 shows that using the successor vector field preserves the expected Bellman velocity target while reducing the variability introduced by sampling a terminal successor displacement. The reduction follows because the successor vector field averages over terminal displacements compatible with the same intermediate successor state.

4.2 QUANTILE-DISTILLED CRITIC

Directly using the flow-based critic for policy extraction requires solving the return-flow ODE, making repeated value queries computationally expensive. Moreover, differentiating the actor objective through the flow critic requires backpropagating through the ODE solver, which can further increase computational cost and lead to unstable optimization. To avoid these difficulties, we distill the return flow into a one-step critic. We further show that, for one-dimensional return flows, each Gaussian source value corresponds to a fixed quantile level of the generated return distribution. This noise–quantile correspondence provides structured supervision for distilling the flow critic into a one-step quantile critic.

Noise–quantile correspondence. A one-dimensional return flow preserves the ordering of its source samples. For a fixed state–action pair (s, a), let $Z_t = \phi^{s,a}_{\theta,t}(\xi)$ with $\xi \sim \mathcal{N}(0, 1)$. Under the standard uniqueness conditions of the ODE, $\phi^{s,a}_{\theta,t}$ is strictly increasing in ξ, and therefore

$$F^{s,a}_t\left(\phi^{s,a}_{\theta,t}(\xi)\right) = \Phi(\xi), \tag{12}$$

where Φ is the standard Gaussian CDF. Thus, although FBM is trained as a continuous vector field, its source noise provides an explicit quantile coordinate for the learned return distribution. Hence the terminal flow directly defines a quantile function

$$q^{\text{flow}}_\theta(s, a, \tau) := \phi^{s,a}_{\theta,1}\big(\Phi^{-1}(\tau)\big), \qquad \tau \in (0, 1). \tag{13}$$

Quantile-preserving distillation. We exploit the exact noise–quantile correspondence to distill the implicit return flow into an explicit one-step quantile critic $q_\psi(s, a, \tau)$. For $(s, a) \sim \mathcal{D}$ and $\tau \sim \text{Unif}(0, 1)$, the target flow provides the corresponding return $y_\tau(s, a) = \phi^{s,a}_{\bar\theta,1}\big(\Phi^{-1}(\tau)\big)$. The one-to-one correspondence between source noise and return quantiles allows y_τ to directly supervise the student at the same quantile level, without constructing Bellman target samples or performing all-pairs quantile regression:

$$\mathcal{L}_{\text{distill}}(\psi) = \mathbb{E}_{(s,a),\tau}\left[\rho^\kappa_\tau(y_\tau(s, a) - q_\psi(s, a, \tau))\right]. \tag{14}$$

Here $\rho^\kappa_\tau(\delta) = |\tau - \mathbf{1}\{\delta < 0\}|\mathcal{H}_\kappa(\delta)$ is the quantile Huber loss, which provides a robust regression objective by smoothing small residuals and limiting the influence of large errors.

In addition, we train the distilled critic with a direct quantile Bellman objective. For successor quantile levels $\{\tau'_j\}_{j=1}^M$, the one-step critic produces successor return samples

$$z'_j = q_{\bar\psi}(s', a', \tau'_j), \qquad \tau'_j \sim \text{Unif}(0, 1),$$

from which we construct the Bellman targets $y^{\text{B}}_j = r + \gamma z'_j$. Given current quantile levels $\{\tau_i\}_{i=1}^N$, we minimize the standard all-pairs quantile regression objective

$$\mathcal{L}_{\text{QR}}(\psi) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\rho^\kappa_{\tau_i}\left(y^{\text{B}}_j - q_\psi(s, a, \tau_i)\right). \tag{15}$$

The same-rank distillation transfers the quantile structure learned by the flow, while the all-pairs quantile regression allows q_ψ to serve as a Bellman-trained distributional critic rather than merely tracking the flow teacher. This additional critic objective reduces the sensitivity of policy optimization to distillation errors by allowing q_ψ to be corrected directly by Bellman targets rather than solely tracking the flow teacher.

4.3 OVERALL ALGORITHM

The complete algorithm consists of three components. We first train the return flow using the flow-time Bellman objective. The resulting flow provides distributional Bellman supervision for the explicit quantile critic, which is trained jointly with the quantile-preserving distillation loss and the quantile Bellman loss:

$$\mathcal{L}_{\text{critic}} = \mathcal{L}_{\text{distill}} + \lambda_{\text{QR}}\mathcal{L}_{\text{QR}}. \tag{16}$$

In parallel, we learn a flow behavior model $\mu_\beta(a \mid s)$ from the offline dataset. The return flow is therefore used to model the return distribution and supervise the explicit critic during training, while policy extraction only requires the distilled critic and the behavior model.

We sample M candidate actions $\{a^{(j)}\}_{j=1}^{M}$ from $\mu_\beta(\cdot \mid s)$ and rank them using the distilled quantile critic. For each candidate action, we draw K Gaussian samples $\xi_k \sim \mathcal{N}(0,1)$ and set $\tau_k = \Phi(\xi_k)$. Its expected return is approximated by

$$\widehat{Q}_\psi(s,a) = \frac{1}{K}\sum_{k=1}^{K} q_\psi(s,a,\tau_k). \tag{17}$$

The final action is selected as

$$a^\star = \arg\max_{j=1,\dots,M} \widehat{Q}_\psi\left(s, a^{(j)}\right), \qquad a^{(j)} \sim \mu_\beta(\cdot \mid s). \tag{18}$$

5 Experiment

The goal of our experiments is to evaluate both the overall offline RL performance of FBM and the mechanisms underlying its critic design. We focus on four questions: **(Q1)** Does FBM improve offline control over strong scalar and distributional critics? **(Q2)** Does flow-time Bellman supervision improve return-distribution learning over endpoint supervision? **(Q3)** How do quantile distillation and the auxiliary quantile Bellman objective affect critic learning and policy extraction? **(Q4)** What is the computational benefit of the distilled critic during action evaluation?

5.1 Offline RL Performance

We first evaluate the complete FBM algorithm on standard offline RL benchmarks. We consider 37 state-based tasks from OGBench and D4RL, including four state-based OGBench domains (`cube-double-play`, `cube-triple-play`, `scene-play`, and `puzzle-4x4-play`) and 12 D4RL Adroit tasks. For OGBench, we report task success rates, while for D4RL we report normalized returns. Unless otherwise specified, results are averaged over four seeds for all tasks. Full environment descriptions and implementation details are provided in Appendix C.

Baselines. We compare FBM with representative scalar, distributional, and flow-based offline RL methods. IQL and FQL provide strong scalar value-based baselines, while IQN and CODAC model return distributions through quantile functions. We additionally compare against Value Flows, which directly models return distributions with a flow-based critic.

Table 1: **Offline evaluation on OGBench and D4RL benchmarks.** We report the aggregated results across environments. Results are averaged over 4 seeds, and bold values indicate means within 95% of the best performance in each row.

Environment	BC	IQL	FQL	ReBRAC	IQN	CODAC	Value Flows	FBM
`cube-double-play` (5 tasks)	3 ± 1	3 ± 0	26 ± 3	0 ± 1	$\mathbf{65 \pm 5}$	52 ± 5	$\mathbf{63 \pm 6}$	$\mathbf{65 \pm 7}$
`cube-triple-play` (5 tasks)	0 ± 0	1 ± 1	6 ± 4	1 ± 1	$\mathbf{12 \pm 2}$	1 ± 1	11 ± 3	$\mathbf{13 \pm 1}$
`scene-play` (5 tasks)	4 ± 2	23 ± 2	$\mathbf{72 \pm 3}$	12 ± 11	59 ± 1	55 ± 4	59 ± 5	59 ± 1
`puzzle-3x3-play` (5 tasks)	2 ± 1	3 ± 3	$\mathbf{97 \pm 2}$	40 ± 10	84 ± 5	24 ± 6	86 ± 5	$\mathbf{96 \pm 2}$
`puzzle-4x4-play` (5 tasks)	0 ± 0	6 ± 4	16 ± 5	12 ± 3	28 ± 4	32 ± 4	18 ± 9	$\mathbf{39 \pm 6}$
`D4RL Adroit` (12 tasks)	25 ± 4	40 ± 2	18 ± 0	$\mathbf{49 \pm 1}$	$\mathbf{51 \pm 2}$	$\mathbf{51 \pm 1}$	47 ± 3	$\mathbf{51 \pm 2}$

Results. Table 1 reports the aggregated performance across offline RL benchmarks, with complete per-taksk results provided in Appendix E. Overall, FBM achieves strong performance across the evaluated domains, improving over existing flow-based and distributional baselines while remaining competitive with strong offline RL methods. On OGBench tasks, FBM obtains consistent gains, particularly on challenging long-horizon manipulation environments where accurate return-distribution modeling is important. On D4RL Adroit tasks, FBM remains competitive with leading approaches, indicating that the proposed flow-based critic and quantile distillation generalize beyond a single benchmark family. These results demonstrate that flow-time Bellman supervision provides a more effective way to train flow-based distributional critics for offline control.

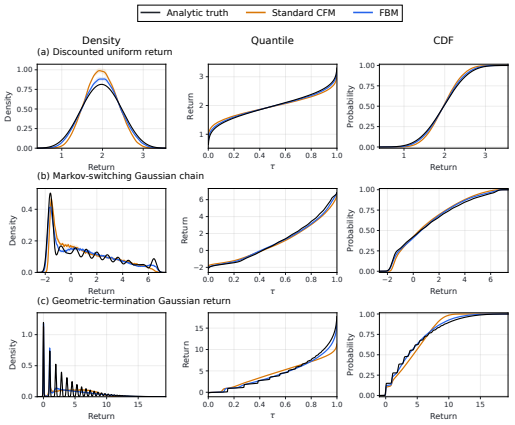

Figure 2: **Return-distribution learning on three representative analytic processes.** FBM more closely matches the analytic density, quantile function, and CDF than STANDARD CFM, particularly for broad and upper-tail return structures.

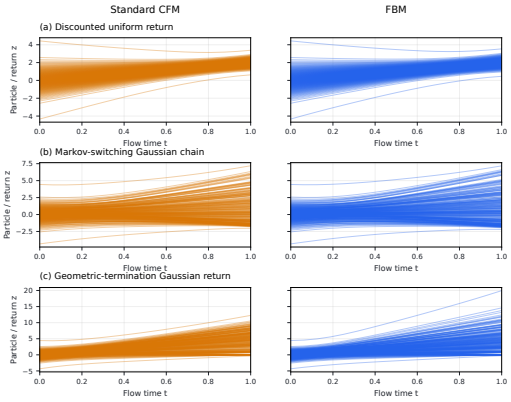

Figure 3: **Learned return-flow trajectories.** FBM captures more expressive return distributions than STANDARD CFM, which tends to produce overly concentrated flow trajectories.

5.2 CONTROLLED ANALYSIS OF FLOW-TIME BELLMAN LEARNING

We isolate the effect of flow-time Bellman supervision using six analytically tractable single-action Markov reward processes covering multimodal, bounded, heavy-tailed, and temporally correlated return structures. We compare FBM with an otherwise identical return flow trained using standard conditional flow matching with terminal Bellman targets, denoted STANDARD-CFM. Full environment definitions and implementation details are provided in Appendix F.

We evaluate the learned return distribution at the initial state against its analytic reference using the 1-Wasserstein distance,

$$W_1(\widehat{Z}, Z) = \int_0^1 \left| F_{\widehat{Z}}^{-1}(\tau) - F_Z^{-1}(\tau) \right| d\tau. \tag{19}$$

We additionally visualize densities, CDFs, and quantile functions to characterize distributional errors.

Results. Table 2 reports the final 1-Wasserstein errors across all six analytic processes. FBM achieves lower mean W_1 than STANDARD-CFM in every case, with a modest improvement on the rare-mode Gaussian process and substantially larger reductions across the remaining return structures. Figure 2 highlights three representative cases. STANDARD-CFM tends to overconcentrate the bounded uniform return and underestimate the upper return range under Markov switching and random termination, whereas FBM more closely matches the analytic densities, quantiles, and CDFs.

Table 2: 1-Wasserstein distance on the analytic toy processes. Mean ± std over five seeds. Lower is better.

Environment	Std.-CFM	FBM
`Rare-mode Gaussian`	0.186 ± 0.079	$\mathbf{0.176 \pm 0.061}$
`Bernoulli Gaussian`	0.066 ± 0.011	$\mathbf{0.036 \pm 0.008}$
`Discounted uniform`	0.071 ± 0.006	$\mathbf{0.031 \pm 0.006}$
`Hypoexponential`	0.209 ± 0.021	$\mathbf{0.109 \pm 0.021}$
`Geometric termination`	0.855 ± 0.040	$\mathbf{0.316 \pm 0.016}$
`Markov switching`	0.265 ± 0.045	$\mathbf{0.150 \pm 0.023}$

Figure 3 provides a complementary view of the learned transport. In the more challenging return processes, STANDARD-CFM concentrates trajectories over a narrower return range, while flow-time Bellman supervision better preserves the spread required by the target distribution. Together, these results show that the improvement of FBM is reflected not only in lower Wasserstein error, but also in more accurate probability allocation across qualitatively different return structures.

5.3 Ablation of the Distilled Quantile Critic

Figure 4 examines the effect of the auxiliary quantile Bellman objective by varying λ_{QR} while keeping all other components fixed. When $\lambda_{\mathrm{QR}} = 0$, the distilled critic is trained purely by matching the flow teacher, which provides a direct transfer of the learned return distribution but does not receive additional Bellman correction. Introducing the quantile Bellman objective consistently improves early-stage learning, with non-zero λ_{QR} values achieving faster performance improvements during training, especially on the more challenging Puzzle 4×4 task. This indicates that Bellman supervision provides an additional learning signal that helps the distilled critic optimize beyond direct teacher matching.

At the same time, the final performance is relatively insensitive to moderate choices of λ_{QR}. On Cube Double, different weights converge to similar success rates, suggesting that the quantile Bellman objective mainly improves optimization efficiency rather than changing the final performance ceiling. However, excessively large weights (e.g., $\lambda_{\mathrm{QR}} = 3$ on Puzzle 4×4) may degrade performance, indicating that overly strong Bellman supervision can interfere with the distributional information transferred from the flow critic. Overall, these results demonstrate that combining quantile distillation with moderate Bellman supervision provides a better balance between preserving the learned return distribution and enabling direct Bellman correction.

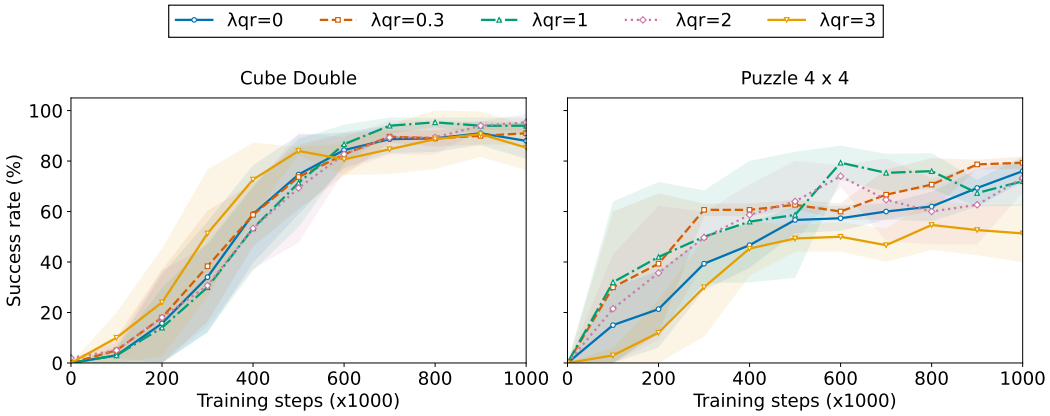

Figure 4: **Effect of the auxiliary quantile Bellman objective.** We vary λ_{QR} while keeping other components fixed. Moderate Bellman supervision improves learning efficiency while maintaining comparable final performance. Shaded regions denote one standard deviation over five seeds.

6 Conclusion

In this work, we introduced *Flow-Time Bellman Matching* (FBM), a framework for distributional reinforcement learning with flow-based critics. Unlike conventional flow-based distributional RL approaches that impose Bellman supervision at the terminal return distribution, FBM derives a flow-time Bellman operator and provides supervision along the entire return generation trajectory. This formulation improves the consistency between flow learning and distributional Bellman backups, leading to more accurate and stable return distribution modeling. Furthermore, we develop a quantile distillation procedure that extracts an efficient one-step critic from the learned return flow, enabling practical policy optimization. Extensive experiments on offline reinforcement learning benchmarks demonstrate that FBM consistently improves performance over existing distributional and flow-based baselines.

AI Use Statement

Generative AI tools were used during the preparation of this work to assist with manuscript writing and editing, brainstorming and organization of research ideas, literature search and organization,

refinement of mathematical exposition, and improvement of figure and presentation quality. The tools were used only as assistive resources throughout the research and writing process. All methodological decisions, theoretical developments, mathematical proofs, experimental implementations, results, citations, and scientific claims were independently developed, verified, and approved by the authors. The authors take full responsibility for the final content of the paper.

Reproducibility Statement

The main paper specifies the core components of FBM, including the flow-time Bellman operator, the corresponding flow-matching training objective, and the quantile-based distillation framework. The appendix provides the complete practical algorithm and numerical integration procedure, proofs of the theoretical results, benchmark and baseline specifications, evaluation protocols, network architectures, hyperparameters, and details of the controlled distribution-learning experiments. The code is available at:`https://anonymous.4open.science/r/fbm-1E7B/README.md`

References

Bhavya Agrawalla, Michal Nauman, Khush Agrawal, and Aviral Kumar. floq: Training critics via flow-matching for scaling compute in value-based rl. In *International Conference on Learning Representations*, 2026.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.

Marc G Bellemare, Will Dabney, and Mark Rowland. Distributional reinforcement learning. MIT Press, 2023.

H. Chen, C. Lu, Z. Wang, H. Su, and J Zhu. Score regularized policy optimization through diffusion behavior. In *International Conference on Learning Representations (ICLR)*, 2024.

Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. *ArXiv*, abs/2209.14548, 2022.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018a.

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018b.

S. Ding, K. Hu, Z. Zhang, K. Ren, W. Zhang, J. Yu, J. Wang, and Y Shi. Diffusion-based reinforcement learning via q-weighted variational policy optimization. In *Neural Information Processing Systems (NeurIPS)*, 2024.

Perry Dong, Chongyi Zheng, Chelsea Finn, Dorsa Sadigh, and Benjamin Eysenbach. Value flows. *ArXiv*, abs/2510.07650, 2025.

Jesse Farebrother, Matteo Pirotta, Andrea Tirinzoni, Rémi Munos, Alessandro Lazaric, and Ahmed Touati. Temporal difference flows. *ArXiv*, abs/2503.09817, 2025.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *ArXiv*, abs/2004.07219, 2021.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *ArXiv*, abs/2106.06860, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning (ICML)*, 2019.

Chen-Xiao Gao, Chenyang Wu, Mingjun Cao, Chenjun Xiao, Yang Yu, and Zongzhang Zhang. Behavior-regularized diffusion policy optimization for offline reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2025.

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

Michael Janner, Igor Mordatch, and Sergey Levine. Generative temporal difference learning for infinite-horizon prediction. *ArXiv*, abs/2010.14496, 2021.

B. Kang, X. Ma, C. Du, T. Pang, and S Yan. Efficient diffusion policies for offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *ArXiv*, abs/2110.06169, 2021.

Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *ArXiv*, abs/1906.00949, 2019.

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

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pp. 45–73. Springer, 2012.

S. Levine, A. Kumar, G. Tucker, and J Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, abs/2005.01643, 2020.

Shiau Hong Lim and Ilyas Malik. Distributional reinforcement learning for risk-sensitive policies. In *Neural Information Processing Systems (NeurIPS)*, 2022.

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

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *ArXiv*, abs/2412.06264, 2024.

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

Xiaoteng Ma, Junyao Chen, Li Xia, Jun Yang, Qianchuan Zhao, and Zhengyuan Zhou. Dsac: Distributional soft actor-critic for risk-sensitive reinforcement learning. *ArXiv*, abs/2004.14547, 2025.

Yecheng Ma, Dinesh Jayaraman, and Osbert Bastani. Conservative offline distributional reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Borislav Mavrin, Shangtong Zhang, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. Distributional reinforcement learning for efficient exploration. *ArXiv*, abs/1905.06125, 2019.

Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2010.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *ArXiv*, abs/2006.09359, 2021.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. *ArXiv*, abs/2410.20092, 2024.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *International Conference on Machine Learning (ICML)*, 2025.

Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 1962.

Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 1956.

Mark Rowland, Marc Bellemare, Will Dabney, Remi Munos, and Yee Whye Teh. An analysis of categorical distributional reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Franki Nguimatsia Tiofack, Théotime Le Hellard, Fabian Schramm, Nicolas Perrin-Gilbert, and Justin Carpentier. Guided flow policy: Learning from high-value actions in offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Kaiwen Wang, Kevin Zhou, Runzhe Wu, Nathan Kallus, and Wen Sun. The benefits of being distributional: Small-loss bounds for reinforcement learning. *ArXiv*, abs/2305.15703, 2023a.

Kaiwen Wang, Owen Oertell, Alekh Agarwal, Nathan Kallus, and Wen Sun. More benefits of being distributional: Second-order bounds for reinforcement learning. *ArXiv*, abs/2402.07198, 2024.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *ArXiv*, abs/2208.06193, 2023b.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *ArXiv*, abs/1911.11361, 2019.

S. Zhang, W. Zhang, and Q Gu. Energy-weighted flow matching for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

A Additional Method Details

This section provides practical details of the flow-time Bellman update, the quantile-distilled critic, and the complete training algorithm.

A.1 Practical Flow-Time Bellman Update

For practical training, we maintain a return-flow critic $v_\theta(z, t, s, a)$ and an exponential-moving-average target critic $v_{\bar{\theta}}$. For each transition $(s, a, r, s', m) \sim \mathcal{D}$, where $m \in \{0, 1\}$ is the continuation mask, we sample a next action $a' \sim \pi_\psi(\cdot \mid s')$, a flow time $t \sim \mathrm{Unif}[0, 1]$, and independent $\xi', \eta \sim \mathcal{N}(0, 1)$. Defining $\widetilde{\gamma} = \gamma m$, we first integrate the target return flow to the sampled flow time,

$$z'_t = \phi_{\bar{\theta},t}^{s',a'}(\xi'). \tag{20}$$

The corresponding flow-time Bellman state and velocity target are

$$x_t = tr + \widetilde{\gamma} z'_t + (1 - t)\sqrt{1 - \widetilde{\gamma}^{\,2}}\,\eta, \tag{21}$$

$$u_t = r + \widetilde{\gamma} v_{\bar{\theta}}(z'_t, t, s', a') - \sqrt{1 - \widetilde{\gamma}^{\,2}}\,\eta. \tag{22}$$

We optimize the return-flow critic using

$$\mathcal{L}_{\mathrm{FBM}}(\theta) = \mathbb{E}\left[\left\|v_\theta(x_t, t, s, a) - \mathrm{sg}(u_t)\right\|_2^2\right], \tag{23}$$

and update the target parameters by

$$\bar{\theta} \leftarrow \rho\bar{\theta} + (1 - \rho)\theta. \tag{24}$$

For terminal transitions, $m = 0$ and hence $\widetilde{\gamma} = 0$, reducing the update to conditional flow matching between Gaussian noise and the immediate reward r.

A.2 QUANTILE-DISTILLED CRITIC

We maintain a one-step quantile critic $q_\psi(s, a, \tau)$. The return flow provides direct supervision through the noise–quantile correspondence established in Section 4.2. For $(s, a) \sim \mathcal{D}$ and $\tau \sim \mathrm{Unif}(0, 1)$, we define the flow teacher target

$$y_\tau^{\mathrm{dist}} = \mathrm{sg}\Big[\phi_{\theta,1}^{s,a}\big(\Phi^{-1}(\tau)\big)\Big]. \tag{25}$$

Because the Gaussian source value $\Phi^{-1}(\tau)$ corresponds directly to the τ-quantile of the generated return distribution, the teacher return directly supervises the student at the same quantile without constructing a Bellman target or performing all-pairs quantile matching. We optimize

$$\mathcal{L}_{\mathrm{distill}}(\psi) = \mathbb{E}_{(s,a),\tau}\left[\rho_\tau^\kappa\big(y_\tau^{\mathrm{dist}} - q_\psi(s, a, \tau)\big)\right]. \tag{26}$$

Here,

$$\rho_\tau^\kappa(\delta) = |\tau - \mathbf{1}\{\delta < 0\}|\,\mathcal{H}_\kappa(\delta), \qquad \mathcal{H}_\kappa(\delta) = \begin{cases} \frac{1}{2}\delta^2, & |\delta| \le \kappa, \\ \kappa\left(|\delta| - \frac{1}{2}\kappa\right), & |\delta| > \kappa, \end{cases} \tag{27}$$

where the Huber component reduces the influence of large distillation residuals.

We additionally train q_ψ with a direct distributional Bellman objective. For a transition $(s, a, r, s', m) \sim \mathcal{D}$, we select the successor action a' using the same candidate-selection rule used for policy extraction and sample successor quantile levels $\{\tau'_j\}_{j=1}^{N'}$ independently from $\mathrm{Unif}(0, 1)$. With $\widetilde{\gamma} = \gamma m$, we construct successor return samples and Bellman targets as

$$z'_j = \mathrm{sg}\big[q_\psi(s', a', \tau'_j)\big], \tag{28}$$

$$y_j^{\mathrm{B}} = r + \widetilde{\gamma} z'_j. \tag{29}$$

For independently sampled current quantile levels $\{\tau_i\}_{i=1}^{N}$, we use the all-pairs quantile regression objective

$$\mathcal{L}_{\mathrm{QR}}(\psi) = \frac{1}{NN'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i}^\kappa\big(y_j^{\mathrm{B}} - q_\psi(s, a, \tau_i)\big). \tag{30}$$

The distilled critic is trained with the combined objective

$$\mathcal{L}_{\mathrm{Q}}(\psi) = \mathcal{L}_{\mathrm{distill}}(\psi) + \lambda_{\mathrm{QR}} \mathcal{L}_{\mathrm{QR}}(\psi). \tag{31}$$

The distillation term transfers the quantile function learned by the return flow through direct same-quantile supervision, while the Bellman term provides an additional bootstrapped learning signal from the one-step critic itself.

A.3 Practical Algorithm

Algorithm 1 Flow-Time Bellman Matching

Input: dataset $\mathcal{D}$, return-flow critic v_θ, quantile critic q_ψ, proposal vector field g_β, candidate budget M, return-sample budget K, coefficient λ_{QR}, training steps N_{train}
Initialize target return-flow critic $v_{\bar\theta} \leftarrow v_\theta$
function SelectAction(s)
  $\{a^{(j)}\}_{j=1}^{M} \sim \mu_\beta(\cdot \mid s)$, $\{\xi_k\}_{k=1}^{K} \sim \mathcal{N}(0,1)$, $\tau_k \leftarrow \Phi(\xi_k)$
  $\widehat{Q}_\psi(s, a^{(j)}) \leftarrow \frac{1}{K}\sum_{k=1}^{K} q_\psi(s, a^{(j)}, \tau_k)$, $j = 1, \dots, M$
  return $a^{(\arg\max_j \widehat{Q}_\psi(s,a^{(j)}))}$
for $n = 1, \dots, N_{\mathrm{train}}$ **do**
  Sample batch $\{(s, a, r, s', m)\} \sim \mathcal{D}$
  $a' \leftarrow$ SelectAction(s'), $\widetilde{\gamma} \leftarrow \gamma m$
  ▷ *Return-flow critic*
  $t \sim \mathrm{Unif}[0,1]$, $\xi', \eta \sim \mathcal{N}(0,1)$
  $z'_t \leftarrow \phi_{\bar\theta,t}^{s',a'}(\xi')$
  $x_t \leftarrow tr + \widetilde{\gamma} z'_t + (1-t)\sqrt{1-\widetilde{\gamma}^2}\,\eta$
  $u_t \leftarrow r + \widetilde{\gamma} v_{\bar\theta}(z'_t, t, s', a') - \sqrt{1-\widetilde{\gamma}^2}\,\eta$
  $\mathcal{L}_{\mathrm{FBM}} \leftarrow \mathbb{E}\Big[\|v_\theta(x_t, t, s, a) - \mathrm{sg}(u_t)\|_2^2\Big]$
  Update θ using $\nabla_\theta \mathcal{L}_{\mathrm{FBM}}$
  ▷ *Quantile-distilled critic*
  $\tau \sim \mathrm{Unif}(0,1)$, $y_\tau^{\mathrm{dist}} \leftarrow \mathrm{sg}\Big[\phi_{\bar\theta,1}^{s,a}\big(\Phi^{-1}(\tau)\big)\Big]$
  $\{\tau_i\}, \{\tau'_j\} \sim \mathrm{Unif}(0,1)$, $y_j^{\mathrm{B}} \leftarrow \mathrm{sg}\big[r + \widetilde{\gamma} q_\psi(s', a', \tau'_j)\big]$
  $\mathcal{L}_{\mathrm{critic}} \leftarrow \mathcal{L}_{\mathrm{distill}} + \lambda_{\mathrm{QR}} \mathcal{L}_{\mathrm{QR}}$
  Update ψ using $\nabla_\psi \mathcal{L}_{\mathrm{critic}}$
  ▷ *Behavior proposal*
  $\zeta \sim \mathcal{N}(0, I)$, $t_a \sim \mathrm{Unif}[0,1]$, $a_{t_a} \leftarrow (1-t_a)\zeta + t_a a$
  $\mathcal{L}_{\mathrm{BC}} \leftarrow \mathbb{E}\Big[\|g_\beta(a_{t_a}, t_a, s) - (a - \zeta)\|_2^2\Big]$
  Update β using $\nabla_\beta \mathcal{L}_{\mathrm{BC}}$
  $\bar\theta \leftarrow \rho\bar\theta + (1-\rho)\theta$
end for
return SelectAction

B Proofs of Main Results

This section proves the four formal results used by the method. The first two results justify the marginal Bellman update and its regression target. The last two establish the noise–quantile correspondence of the learned one-dimensional flow and the variance comparison for randomized quantile stratification. Throughout, all Gaussian variables introduced in a proof are mutually independent and independent of the transition variables unless stated otherwise.

B.1 Consistency of the Flow-Time Bellman Operator

For return distributions $Z_t^{(1)}$ and $Z_t^{(2)}$ at flow time t, define the maximal p-Wasserstein metric

$$\bar{d}_p\left(Z_t^{(1)}, Z_t^{(2)}\right) := \sup_{s,a} W_p\left(Z_t^{(1)}(s,a), Z_t^{(2)}(s,a)\right). \tag{32}$$

Proof of Proposition 1. Fix an arbitrary $t \in [0,1]$. For a flow-time return distribution Z_t, let

$$(\mathsf{P}^\pi Z_t)(s,a)$$

denote the successor-return distribution obtained by first sampling $S' \sim p(\cdot \mid s,a)$, $A' \sim \pi(\cdot \mid S')$, and then $z'_t \sim Z_t(S', A')$. The flow-time Bellman operator is therefore

$$(\mathcal{B}_t^\pi Z_t)(s,a) = \mathrm{Law}\left(tr(s,a) + \gamma z'_t + \sigma_t \eta\right), \qquad \sigma_t := (1-t)\sqrt{1-\gamma^2}, \tag{33}$$

where $\eta \sim \mathcal{N}(0,1)$ is independent.

Consider two arbitrary flow-time return distributions $Z_t^{(1)}$ and $Z_t^{(2)}$. The successor-mixture operator is non-expansive in maximal Wasserstein distance:

$$\begin{aligned} & W_p^p\left((\mathsf{P}^\pi Z_t^{(1)})(s,a), (\mathsf{P}^\pi Z_t^{(2)})(s,a)\right) \\ & \quad \le \mathbb{E}_{S',A'}\left[W_p^p\left(Z_t^{(1)}(S',A'), Z_t^{(2)}(S',A')\right)\right] \\ & \quad \le \bar{d}_p^p\left(Z_t^{(1)}, Z_t^{(2)}\right). \end{aligned} \tag{34}$$

Translation leaves the Wasserstein distance unchanged, scaling by γ scales it by γ, and convolution with the common Gaussian law $\mathcal{N}(0,\sigma_t^2)$ is non-expansive. Hence,

$$\begin{aligned} & W_p\left((\mathcal{B}_t^\pi Z_t^{(1)})(s,a), (\mathcal{B}_t^\pi Z_t^{(2)})(s,a)\right) \\ & \quad \le \gamma W_p\left((\mathsf{P}^\pi Z_t^{(1)})(s,a), (\mathsf{P}^\pi Z_t^{(2)})(s,a)\right) \\ & \quad \le \gamma \bar{d}_p\left(Z_t^{(1)}, Z_t^{(2)}\right). \end{aligned} \tag{35}$$

Taking the supremum over (s,a) gives

$$\bar{d}_p\left(\mathcal{B}_t^\pi Z_t^{(1)}, \mathcal{B}_t^\pi Z_t^{(2)}\right) \le \gamma \bar{d}_p\left(Z_t^{(1)}, Z_t^{(2)}\right). \tag{36}$$

Since t was arbitrary, Eq. (36) holds for every $t \in [0,1]$. □

B.2 Consistency of Bellman Flow Matching

Proof of Proposition 2. Fix a Bellman iteration n and a state–action pair (s,a), and let

$$p_t^{(n+1)} := \mathcal{B}_t^\pi p_t^{(n)}. \tag{37}$$

Let $Y_t^{(n)}$ be a successor sample generated by the vector field $v^{(n)}$,

$$\frac{d}{dt} Y_t^{(n)} = v^{(n)}\left(Y_t^{(n)}, t, S', A'\right), \tag{38}$$

and define

$$X_t^{(n+1)} = tr + \gamma Y_t^{(n)} + (1-t)\sqrt{1-\gamma^2}\,\eta. \tag{39}$$

By construction,

$$\mathrm{Law}\left(X_t^{(n+1)} \mid s,a\right) = p_t^{(n+1)}(\cdot \mid s,a). \tag{40}$$

Differentiating Eq. (39) gives the sample-level velocity

$$U_t^{(n+1)} = r + \gamma v^{(n)}\left(Y_t^{(n)}, t, S', A'\right) - \sqrt{1-\gamma^2}\,\eta. \tag{41}$$

Consider the population regression objective

$$\mathcal{L}(g) = \mathbb{E}\left[\left|g(X_t^{(n+1)}, t, S, A) - U_t^{(n+1)}\right|^2\right]. \tag{42}$$

By the standard L_2 projection property, its population minimizer is

$$v^{(n+1),\star}(x,t,s,a) = \mathbb{E}\left[U_t^{(n+1)} \mid X_t^{(n+1)} = x, S = s, A = a\right]. \tag{43}$$

Thus Eq. (11) recovers the posterior-averaged velocity of the Bellman sample paths.

By the standard marginalization result for flow matching (Lipman et al., 2023), the vector field in Eq. (43) generates the marginal probability path of $X_t^{(n+1)}$. Therefore, if $\phi_t^{(n+1)}$ denotes the flow induced by $v^{(n+1),\star}$, then

$$\left(\phi_t^{(n+1)}\right)_{\#}\mathcal{N}(0,1) = p_t^{(n+1)} = \mathcal{B}_t^\pi p_t^{(n)}. \tag{44}$$

Hence the population-optimal Bellman flow-matching objective recovers a vector field whose ODE generates the exact flow-time Bellman-updated return distribution. □

B.3 Variance Reduction of Bellman Velocity Targets

Proof of Proposition 3. Let

$$M_t := v^{\star}_{\mathrm{suc}}(Z'_t, t, H) = \mathbb{E}[D' \mid Z'_t, H].$$

Then

$$U_t^{\mathrm{term}} = r(s,a) + \gamma D' - \sqrt{1-\gamma^2}\eta,$$

whereas

$$U_t^{\mathrm{flow}} = r(s,a) + \gamma M_t - \sqrt{1-\gamma^2}\eta.$$

Taking expectations of $M_t = \mathbb{E}[D' \mid Z'_t, H]$ gives

$$\mathbb{E}[M_t \mid s,a] = \mathbb{E}[D' \mid s,a],$$

and therefore

$$\mathbb{E}[U_t^{\mathrm{flow}} \mid s,a] = \mathbb{E}[U_t^{\mathrm{term}} \mid s,a].$$

Since η is independent of the successor variables and appears identically in both targets,

$$\begin{aligned}\mathrm{Var}\left(U_t^{\mathrm{term}} \mid s,a\right) - \mathrm{Var}\left(U_t^{\mathrm{flow}} \mid s,a\right) \\ = \gamma^2\left[\mathrm{Var}(D' \mid s,a) - \mathrm{Var}(M_t \mid s,a)\right].\end{aligned} \tag{45}$$

By the conditional variance decomposition,

$$\mathrm{Var}(D' \mid s,a) = \mathrm{Var}(M_t \mid s,a) + \mathbb{E}\left[\mathrm{Var}\left(D' \mid Z'_t, H\right) \mid s,a\right].$$

Substituting into Eq. (45) yields

$$\begin{aligned}\mathrm{Var}\left(U_t^{\mathrm{term}} \mid s,a\right) - \mathrm{Var}\left(U_t^{\mathrm{flow}} \mid s,a\right) \\ = \gamma^2\mathbb{E}\left[\mathrm{Var}\left(D' \mid Z'_t, S', A'\right) \mid s,a\right] \geq 0.\end{aligned}$$

□

B.4 Noise–Quantile Correspondence

The next proof applies only to the learned one-dimensional marginal flow. It does not assign a global quantile rank to an arbitrary realization of the hidden Bellman coupling.

Proof of the noise–quantile correspondence. Fix (s,a) and a flow time t. Consider two initial noises $\xi_1 < \xi_2$. If their trajectories intersected at some time $t_\star$, then

$$\phi^{s,a}_{\theta,t_\star}(\xi_1) = \phi^{s,a}_{\theta,t_\star}(\xi_2). \tag{46}$$

Local Lipschitz continuity of the vector field gives uniqueness of the ODE solution through this common point, both forward and backward on the interval of existence. The two trajectories would therefore have the same initial value, contradicting $\xi_1 < \xi_2$. Hence

$$\xi_1 < \xi_2 \quad \Longrightarrow \quad \phi^{s,a}_{\theta,t}(\xi_1) < \phi^{s,a}_{\theta,t}(\xi_2), \tag{47}$$

so the scalar flow map is strictly increasing.

Let $\Xi \sim \mathcal{N}(0,1)$ and $Z_t = \phi^{s,a}_{\theta,t}(\Xi)$. Strict monotonicity implies, for every $x \in \mathbb{R}$,

$$\begin{aligned}F^{s,a}_{\theta,t}\left(\phi^{s,a}_{\theta,t}(x)\right) &= \mathbb{P}\left(\phi^{s,a}_{\theta,t}(\Xi) \leq \phi^{s,a}_{\theta,t}(x)\right) \\ &= \mathbb{P}(\Xi \leq x) = \Phi(x).\end{aligned} \tag{48}$$

Substituting the random input $x = \Xi$ proves Eq. (12). Finally, set $x = \Phi^{-1}(\tau)$. Equation (48) shows that $\phi^{s,a}_{\theta,t}(\Phi^{-1}(\tau))$ has CDF value τ. Since a continuous strictly increasing transformation of a Gaussian variable has a continuous CDF on its support, this value is the τ-quantile, proving Eq. (13). □

C EXPERIMENTAL DETAILS

This section describes the environments, datasets, baselines, evaluation metrics, and compute used in our experiments.

C.1 BENCHMARKS AND DATASETS

We evaluate FBM on 42 state-based offline RL tasks from OGBench and D4RL. Throughout the experiments, we train on fixed offline datasets consisting of transition tuples $\mathcal{D} = \{(s_i, a_i, r_i, s'_i, m_i)\}_{i=1}^{N}$, where $m_i \in \{0, 1\}$ denotes the continuation mask.

OGBench. We use the reward-based single-task variants of OGBench for standard offline RL. Our evaluation includes six state-based environments: `antmaze-large-navigate`, `cube-double-play`, `cube-triple-play`, `scene-play`, `puzzle-3x3-play`, and `puzzle-4x4-play`. Each environment provides five separate single-task variants, yielding 30 OGBench tasks in total. The manipulation environments use play-style datasets containing diverse and temporally extended behaviors, while `antmaze-large-navigate` evaluates long-horizon locomotion from offline navigation data. Performance is reported as task success rate following the standard OGBench evaluation protocol.

D4RL. We additionally evaluate on the 12 Adroit tasks from D4RL, which involve high-dimensional dexterous manipulation with a 24-DoF robotic hand. We consider the `human`, `cloned`, and `expert` datasets for each of the four environments `pen`, `door`, `hammer`, and `relocate`, giving 12 tasks in total. Following the standard D4RL protocol, we report normalized returns for these tasks.

C.2 BASELINES

We compare against four groups of baselines. Our organization separates the effect of the policy class from the effect of the critic representation.

Scalar critics with Gaussian policies. We first compare to standard offline RL methods that learn scalar value functions and use Gaussian actors. **BC** trains a Gaussian policy by behavior cloning. **IQL** (Kostrikov et al., 2021) learns scalar value functions with expectile regression and extracts the policy by advantage-weighted regression. **ReBRAC** (Tarasov et al., 2023) is a strong TD3+BC-style offline actor-critic method with additional regularization and normalization choices. These baselines provide representative scalar-value offline RL references for assessing the performance of standard value-based methods.

Scalar critics with flow policies. We next compare to methods that keep the critic scalar but use expressive flow-based policies. **FBRAC** combines behavior-regularized actor-critic learning with a flow policy. **FQL** (Park et al., 2025) trains a one-step flow actor to maximize a learned scalar critic while regularizing the actor toward a behavior flow. These baselines control for the possibility that performance gains come mainly from the expressive actor rather than from the proposed distributional critic.

Classical distributional critics. We also compare to distributional RL methods with quantile-based return parameterizations. **IQN** (Dabney et al., 2018b) represents the return distribution through sampled quantile functions. **CODAC** (Ma et al., 2021) combines quantile regression with conservative constraints for offline distributional RL. For these methods, we use the same flow-policy extraction scheme where applicable, so that the comparison focuses on the return representation and critic objective.

Flow-based distributional baseline. Finally, we include **Value Flows** (Dong et al., 2025) as the closest baseline. Value Flows models the return distribution with a flow-based critic and extracts scalar action values from the learned return flow. This baseline is especially important because it uses a continuous generative critic rather than a finite set of learned quantiles. The comparison to Value Flows isolates the contribution of flow-time Bellman supervision beyond the use of a generative return critic.

C.3 Evaluation Protocol

OGBench. For all OGBench tasks, agents are trained for 10^6 gradient steps and evaluated every 10^5 steps. Following the standard OGBench reporting scheme, we report the average success rate over the final three evaluation checkpoints at 800k, 900k, and 1M steps. Success rate is defined as the fraction of evaluation episodes that successfully complete the task. Each checkpoint is evaluated over 100 episodes, and we report the mean and standard deviation over 4 random seeds. For environment-level results, we average the task-level success rates over the five single-task variants within each environment.

D4RL. For D4RL Adroit tasks, agents are trained for 5×10^5 gradient steps and evaluated at the final checkpoint. We report normalized returns following the standard D4RL evaluation protocol.

Implementation and model selection. Our method and all re-run baselines are implemented in JAX using matched training budgets and evaluation schedules, with comparable network architectures where applicable. Hyperparameters are selected using separate tuning seeds, and the seeds used for final evaluation are not used for hyperparameter selection.

Compute and software. Each run is executed on a single NVIDIA GeForce RTX 4090 GPU and takes approximately 8 hours. Experiments are implemented in JAX 0.6.2 with jaxlib 0.6.2 and MuJoCo 3.9.0, using CUDA 12.8 on a workstation equipped with an Intel Core i9-14900K CPU.

D Hyperparameters

We use the same optimization and network architecture settings across all tasks unless otherwise specified. Following prior work (Park et al., 2025; Dong et al., 2025), task-dependent hyperparameters are selected at the environment level for OGBench and reused across the five corresponding single-task variants. For D4RL Adroit, task-specific settings are used. Tables 3 and 4 summarize the shared and task-dependent configurations, respectively.

Table 3: **Common hyperparameters for FBM and re-run baselines.**

Hyperparameter	Value
Optimizer	Adam
Learning rate	3×10^{-4}
Batch size	256
MLP hidden dimensions	$[512, 512, 512, 512]$
Activation	GELU
Value layer normalization	yes
Flow integration steps	10
Candidate actions	16
Target update coefficient	0.005
Number of critic ensembles	2
Quantiles per action (FBM)	16
Quantile Huber threshold κ (FBM)	0.9
Quantile Bellman coefficient λ_{QR} (FBM)	1.0
Flow-time sampling (FBM)	$t \sim \text{Unif}[0, 1]$

Table 4: **Task-specific hyperparameters for FBM and baselines.** For OGBench, the configuration selected for each domain is reused across its five single-task variants. The discount factor γ is shared across methods within each domain. "–" denotes a setting that is not applicable or a result taken directly from prior work.

	Discount	IQL	ReBRAC		FBRAC	IFQL	FQL	C51	IQN	CODAC		Value Flows		FBM
Domain or task	γ	α	α_1	α_2	α	N	α	N	κ	κ	α	λ	τ	λ_{QR}
cube-double-play	0.995	0.3	0.1	0	100	32	100	101	0.9	0.95	300	1	3	1
cube-triple-play	0.995	10	0.03	0	100	32	300	51	0.8	0.95	1000	3	0.03	1
puzzle-3x3-play	0.99	10	0.3	0.001	100	32	1000	101	0.8	0.95	100	0.5	0.3	1
puzzle-4x4-play	0.99	3	0.3	0.01	300	32	1000	101	0.95	0.95	1000	3	100	1
scene-play	0.99	10	0.1	0.001	100	32	300	51	0.95	0.95	100	1	0.3	1
pen-human-v1	0.99	–	–	–	30000	32	10000	51	0.8	0.8	10000	3	1	3
pen-cloned-v1	0.99	–	–	–	10000	32	10000	51	0.8	0.8	10000	3	1	3
pen-expert-v1	0.99	–	–	–	30000	32	3000	51	0.8	0.8	10000	3	0.01	3
door-human-v1	0.99	–	–	–	30000	32	30000	101	0.9	0.9	10000	3	0.01	3
door-cloned-v1	0.99	–	–	–	10000	128	30000	51	0.9	0.9	30000	10	0.3	3
door-expert-v1	0.99	–	–	–	30000	32	30000	51	0.9	0.9	10000	10	0.3	3
hammer-human-v1	0.99	–	–	–	30000	32	30000	51	0.7	0.8	30000	3	0.3	3
hammer-cloned-v1	0.99	–	–	–	10000	32	10000	51	0.7	0.8	10000	3	0.3	3
hammer-expert-v1	0.99	–	–	–	30000	32	30000	51	0.9	0.8	10000	10	1	3
relocate-human-v1	0.99	–	–	–	30000	128	10000	101	0.9	0.9	30000	10	0.01	3
relocate-cloned-v1	0.99	–	–	–	3000	32	30000	51	0.9	0.9	30000	3	0.01	3
relocate-expert-v1	0.99	–	–	–	30000	32	30000	101	0.9	0.9	10000	3	0.1	3

E Additional Experimental Results

Table 5: **Offline RL results with additional baselines.** OGBench reports success rates (%), and D4RL reports normalized returns. Values are mean ± standard deviation. Supplied-data values are shown in black. Bold entries indicate means at or above 95% of the row maximum.

Task	BC	IQL	FQL	IFQL	ReBRAC	C51	IQN	CODAC	Value Flows	FBM
cube-double-play-singletask-task1-v0	13 ± 4	17 ± 1	44 ± 3	28 ± 11	2 ± 3	8 ± 2	82 ± 2	64 ± 4	**90 ± 2**	**94 ± 3**
cube-double-play-singletask-task2-v0	0 ± 0	0 ± 0	42 ± 6	5 ± 1	0 ± 0	0 ± 1	61 ± 12	50 ± 9	**81 ± 15**	**82 ± 4**
cube-double-play-singletask-task3-v0	0 ± 0	0 ± 0	9 ± 4	5 ± 1	0 ± 0	0 ± 0	**72 ± 7**	57 ± 3	65 ± 5	56 ± 4
cube-double-play-singletask-task4-v0	0 ± 0	0 ± 0	2 ± 0	0 ± 0	0 ± 0	0 ± 0	**32 ± 4**	9 ± 6	11 ± 4	12 ± 13
cube-double-play-singletask-task5-v0	0 ± 0	0 ± 0	31 ± 1	18 ± 3	0 ± 0	1 ± 1	**79 ± 2**	**81 ± 2**	69 ± 5	**80 ± 11**
cube-triple-play-singletask-task1-v0	0 ± 0	6 ± 5	28 ± 19	38 ± 10	3 ± 3	0 ± 1	**59 ± 8**	3 ± 3	54 ± 13	**59 ± 3**
cube-triple-play-singletask-task2-v0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	**2 ± 3**	0 ± 1	0 ± 0	0 ± 0
cube-triple-play-singletask-task3-v0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 1	1 ± 1	2 ± 2	**4 ± 1**
cube-triple-play-singletask-task4-v0	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**
cube-triple-play-singletask-task5-v0	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**
scene-play-singletask-task1-v0	19 ± 4	80 ± 0	**99 ± 1**	**98 ± 3**	33 ± 41	19 ± 7	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**
scene-play-singletask-task2-v0	2 ± 3	9 ± 1	84 ± 14	79 ± 13	21 ± 7	2 ± 2	**99 ± 1**	90 ± 8	90 ± 18	**99 ± 1**
scene-play-singletask-task3-v0	0 ± 0	21 ± 7	**99 ± 1**	30 ± 11	5 ± 7	4 ± 3	**97 ± 2**	85 ± 13	**97 ± 3**	**96 ± 6**
scene-play-singletask-task4-v0	1 ± 1	2 ± 0	**80 ± 0**	0 ± 0	0 ± 0	1 ± 1	0 ± 0	0 ± 0	10 ± 5	0 ± 0
scene-play-singletask-task5-v0	0 ± 0	**3 ± 4**	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-3x3-play-singletask-task1-v0	6 ± 3	8 ± 8	**100 ± 0**	94 ± 3	**96 ± 3**	5 ± 2	**99 ± 2**	92 ± 5	**100 ± 0**	**100 ± 0**
puzzle-3x3-play-singletask-task2-v0	0 ± 0	2 ± 3	**97 ± 4**	36 ± 11	14 ± 14	3 ± 3	86 ± 9	4 ± 7	**100 ± 0**	**100 ± 0**
puzzle-3x3-play-singletask-task3-v0	1 ± 1	1 ± 1	92 ± 0	0 ± 0	38 ± 20	1 ± 1	**96 ± 2**	13 ± 13	92 ± 7	**100 ± 0**
puzzle-3x3-play-singletask-task4-v0	0 ± 0	1 ± 1	**96 ± 3**	4 ± 3	15 ± 1	0 ± 0	41 ± 9	9 ± 7	94 ± 4	**99 ± 1**
puzzle-3x3-play-singletask-task5-v0	2 ± 3	1 ± 1	**98 ± 3**	64 ± 6	36 ± 14	2 ± 2	**97 ± 3**	0 ± 0	45 ± 13	81 ± 9
puzzle-4x4-play-singletask-task1-v0	1 ± 1	8 ± 6	21 ± 16	55 ± 16	19 ± 1	0 ± 0	51 ± 11	60 ± 3	19 ± 5	**74 ± 2**
puzzle-4x4-play-singletask-task2-v0	0 ± 0	7 ± 1	11 ± 7	3 ± 1	12 ± 0	0 ± 0	14 ± 3	**26 ± 4**	**27 ± 3**	16 ± 1
puzzle-4x4-play-singletask-task3-v0	0 ± 0	5 ± 4	22 ± 0	47 ± 4	14 ± 6	0 ± 0	33 ± 4	39 ± 8	14 ± 20	**58 ± 3**
puzzle-4x4-play-singletask-task4-v0	0 ± 0	8 ± 8	17 ± 1	18 ± 6	6 ± 3	0 ± 0	24 ± 3	21 ± 3	19 ± 7	**30 ± 11**
puzzle-4x4-play-singletask-task5-v0	0 ± 0	0 ± 0	8 ± 3	2 ± 0	8 ± 3	0 ± 0	**16 ± 0**	14 ± 2	13 ± 11	**16 ± 11**
pen-human-v1	14 ± 0	32 ± 6	0 ± 0	45 ± 1	**69 ± 3**	**69 ± 8**	**68 ± 3**	59 ± 1	**66 ± 4**	**69 ± 4**
pen-cloned-v1	51 ± 2	46 ± 14	1 ± 1	67 ± 10	47 ± 7	61 ± 8	**79 ± 7**	66 ± 1	73 ± 10	**79 ± 2**
pen-expert-v1	89 ± 10	96 ± 3	2 ± 3	92 ± 3	129 ± 4	109 ± 2	125 ± 3	**139 ± 7**	115 ± 4	115 ± 3
door-human-v1	0 ± 0	0 ± 0	0 ± 0	3 ± 4	0 ± 0	0 ± 0	0 ± 0	2 ± 1	**7 ± 1**	**7 ± 2**
door-cloned-v1	0 ± 0	0 ± 0	0 ± 0	**1 ± 1**	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
door-expert-v1	48 ± 13	**104 ± 1**	**104 ± 0**	99 ± 1	**105 ± 2**	**104 ± 0**	**105 ± 0**	**105 ± 1**	**105 ± 1**	**105 ± 1**
hammer-human-v1	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	2 ± 1	2 ± 1	**3 ± 1**	1 ± 0	2 ± 1
hammer-cloned-v1	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	**3 ± 0**	1 ± 0	**3 ± 0**
hammer-expert-v1	89 ± 16	99 ± 1	0 ± 0	84 ± 6	**134 ± 0**	95 ± 15	123 ± 3	**128 ± 3**	100 ± 11	126 ± 5
relocate-human-v1	0 ± 0	0 ± 0	0 ± 0	**3 ± 1**	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	1 ± 1
relocate-cloned-v1	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**
relocate-expert-v1	6 ± 4	99 ± 1	**106 ± 1**	98 ± 16	**106 ± 1**	**105 ± 1**	**107 ± 1**	**103 ± 1**	101 ± 3	**107 ± 4**

F Toy Experiments: Learning Analytic Return Distributions

These experiments isolate how Bellman-consistent return-flow learning changes distribution recovery relative to endpoint conditional flow matching (CFM). Six analytically tractable Markov reward processes cover separated modes, independent reward events, bounded returns, skewed tails, random termination, and persistent reward regimes. Each process is an MDP with one available action, so

policy learning and action selection cannot explain the differences. We evaluate the return distribution at the initial state using density estimates, cumulative distribution functions (CDFs), quantiles, and Wasserstein distance. Density recovery depends on both the learned distribution and the estimator used to visualize it. We use Gaussian kernel density estimation (KDE) with different bandwidths to evaluate the recovered densities (Rosenblatt, 1956; Parzen, 1962).

F.1 Controlled Comparison and Evaluation

Training objectives. Both methods learn a scalar velocity field conditioned on state and flow time, with independent standard Gaussian source variables ξ and ξ'. For a sampled transition (s, r, s', m), write $d = \gamma m$, where $m = 0$ on termination. CFM first samples the successor endpoint $Y_1 = \phi^{s'}_{\bar{\theta},1}(\xi')$, constructs $B = r + dY_1$, and regresses the velocity at $X_t = (1-t)\xi + tB$ toward $B - \xi$. FBM instead samples the intermediate successor $Y_t = \phi^{s'}_{\bar{\theta},t}(\xi')$ and uses $X_t = tr + dY_t + (1-t)\sqrt{1-d^2}\xi$ with target velocity $U_t = r + d\, v_{\bar{\theta}}(Y_t, t, s') - \sqrt{1-d^2}\xi$. Both minimize mean squared velocity error with $t \sim \mathrm{Unif}(0, 1)$.

Shared protocol. Each toy uses five independently parameterized models per method (seeds 0–4), trained for 100,000 updates with batches of 256 freshly sampled transitions. The velocity network concatenates scalar return, flow time, and a 40-dimensional padded one-hot state encoding, followed by three width-128 SiLU hidden layers and a scalar output. Both methods use Adam with learning rate 3×10^{-4} and target update $\bar{\theta} \leftarrow 0.99\bar{\theta} + 0.01\theta$. Paired runs share initialization, transition batches, flow times, and Gaussian draws. Evaluation uses the final EMA critic with 16 Euler steps and 20,000 Gaussian inputs shared between methods within each seed. Learning curves use 4,096 fixed inputs every 10,000 updates.

Distribution diagnostics. For each environment, we compare the learned return distribution at the initial state with the analytic reference distribution using CDFs, quantile functions, and Wasserstein distance. For the Bernoulli Gaussian, geometric-termination, and Markov-switching environments, the reference distribution is the policy-marginal distribution under the fixed uniform policy, obtained by averaging the two action-conditional distributions. Learned distributions are estimated from generated samples, while reference quantiles are computed from the analytic CDF. We report quantile errors $E_\tau = |\hat{q}(\tau) - q_*(\tau)|$ at $\tau \in \{0.01, 0.10, 0.50, 0.90, 0.99\}$ and estimate W_1 from empirical quantile differences. Density plots use Gaussian-smoothed histograms with the same bandwidth for both methods. All curves show the mean and standard deviation over five seeds.

F.2 Analytically Tractable Toy Environments

We consider a set of analytically tractable toy environments with known return distributions. The two-action environments induce Markov reward processes (MRPs) under the fixed uniform policy $\pi(a \mid s) = 1/2$. These environments provide analytic references for evaluating distributional approximation errors, with numerical quantile inversion and a truncated mixture for geometric termination. The reported Wasserstein distances use reference samples drawn from these distributions. For each environment, we compare the reference return distribution with the learned distribution; the two-action results shown here marginalize over the initial action under the fixed policy.

The first environment is a rare-mode Gaussian chain with horizon $H = 20$ and discount factor $\gamma = 0.95$. At the beginning of each episode, a latent variable $C \sim \mathrm{Bernoulli}(0.2)$ is sampled and remains fixed throughout the trajectory. Rewards are generated as $R_j = \mu_C + \sigma\epsilon_j$, where $\mu_0 = 0$, $\mu_1 = 0.5$, $\sigma = 0.05$, and $\epsilon_j \sim \mathcal{N}(0, 1)$ independently. The resulting return distribution is a Gaussian mixture with weights 0.8 and 0.2:

$$Z \mid C = c \sim \mathcal{N}\left(\mu_c \sum_{j=0}^{H-1} \gamma^j, \sigma^2 \sum_{j=0}^{H-1} \gamma^{2j}\right). \tag{49}$$

The Bernoulli Gaussian environment consists of a finite-horizon chain with $H = 8$, $\gamma = 0.7$, and two actions. At each step, the reward is given by $R_j = C_j + \sigma\epsilon_j$, where $\sigma = 0.05$ and $C_j \mid a_j \sim \mathrm{Bernoulli}(p_{a_j})$, with $p_0 = 0.1$ and $p_1 = 0.3$. Under the fixed uniform policy, the return

distribution is obtained by marginalizing over all binary reward sequences:

$$Z \mid c \sim \mathcal{N}\left(\sum_{j=0}^{H-1} \gamma^j c_j, \sigma^2 \sum_{j=0}^{H-1} \gamma^{2j}\right), \tag{50}$$

where $c \in \{0,1\}^H$ has weight $w_c = 0.2^{|c|}0.8^{H-|c|}$ and $|c| = \sum_j c_j$. Conditional on the initial action a, the weight becomes $w_c^a = p_a^{c_0}(1-p_a)^{1-c_0}\prod_{j=1}^{H-1} 0.2^{c_j}0.8^{1-c_j}$.

The discounted uniform MRP contains independent rewards $R_j \sim \mathrm{Unif}(0,1)$ with horizon $H = 7$ and discount factor $\gamma = 0.8$. The return is

$$Z = \sum_{j=0}^{H-1} \gamma^j R_j, \tag{51}$$

whose distribution is given by the weighted sum of independent uniform random variables.

The hypoexponential MRP uses independent exponential rewards $R_j \sim \mathrm{Exp}(1)$ with horizon $H = 8$ and $\gamma = 0.75$, where the exponential distribution is parameterized by its rate. Since $\gamma^j R_j \sim \mathrm{Exp}(\gamma^{-j})$, the return distribution follows a hypoexponential distribution:

$$F_Z(z) = 1 - \sum_{j=0}^{H-1} c_j e^{-\lambda_j z}, \qquad z \geq 0, \tag{52}$$

with $F_Z(z) = 0$ for $z < 0$, where $\lambda_j = \gamma^{-j}$ and

$$c_j = \prod_{k \neq j} \frac{\lambda_k}{\lambda_k - \lambda_j}. \tag{53}$$

The geometric-termination environment contains an active state and an absorbing state, with $\gamma = 0.95$, $\sigma = 0.05$, and two actions. At each active-state transition, action a terminates the process with probability q_a, where $q_0 = 0.1$ and $q_1 = 0.2$, and emits the terminal reward $R = \sigma\epsilon$; otherwise, it receives reward $R = 1 + \sigma\epsilon$ and continues. Let L denote the number of nonterminal transitions before termination, including $L = 0$. Conditioning on L, the return distribution is Gaussian:

$$Z \mid L = l \sim \mathcal{N}\left(\sum_{j=0}^{l-1} \gamma^j, \sigma^2 \sum_{j=0}^{l} \gamma^{2j}\right). \tag{54}$$

Under the fixed uniform policy, $\Pr(L = l) = 0.15(0.85)^l$. Conditional on the initial action a, the weights are $w_0^a = q_a$ and $w_l^a = (1-q_a)(0.85)^{l-1}0.15$ for $l \geq 1$. Each action-conditional reference retains $l = 0, \ldots, 113$ and renormalizes its weights, with omitted mass below 10^{-8}; the plotted reference averages the two resulting distributions.

Finally, the Markov-switching Gaussian environment introduces a regime variable $C_j \in \{0,1\}$, with $H = 10$ and $\gamma = 0.95$. The root draws C_0 from $\rho = (2/3, 1/3)$, and subsequent states encode time and regime. Rewards are generated as $R_j = \mu_{C_j} + \sigma\epsilon_j$, with $(\mu_0, \mu_1) = (-0.2, 0.8)$ and $\sigma = 0.08$. After the reward, action a_j selects the next-regime transition matrix

$$P^{(0)} = \begin{pmatrix} 0.95 & 0.05 \\ 0.30 & 0.70 \end{pmatrix}, \qquad P^{(1)} = \begin{pmatrix} 0.85 & 0.15 \\ 0.10 & 0.90 \end{pmatrix}. \tag{55}$$

Under the fixed uniform policy, write $\bar{P} = (P^{(0)} + P^{(1)})/2$. Marginalizing over all regime sequences yields a Gaussian mixture with weights $w_c = \rho_{c_0}\prod_{j=1}^{H-1} \bar{P}_{c_{j-1},c_j}$ and component means

$$m_c = \sum_{j=0}^{H-1} \gamma^j \mu_{c_j}, \tag{56}$$

and shared variance

$$\sigma_Z^2 = \sigma^2 \sum_{j=0}^{H-1} \gamma^{2j}. \tag{57}$$

Conditional on the initial action a, the weights become $w_c^a = \rho_{c_0} P^{(a)}_{c_0,c_1} \prod_{j=2}^{H-1} \bar{P}_{c_{j-1},c_j}$.

Table 6: Return-distribution errors on toy benchmarks.

Toy	Method	W_1	$E_{.01}$	$E_{.10}$	$E_{.50}$	$E_{.90}$	$E_{.99}$
Rare-mode Gaussian	CFM	0.186 ± 0.079	0.169 ± 0.029	0.103 ± 0.015	0.026 ± 0.012	0.052 ± 0.028	0.066 ± 0.027
	FBM	0.176 ± 0.061	0.046 ± 0.027	0.035 ± 0.008	0.010 ± 0.008	0.076 ± 0.006	0.023 ± 0.014
Bernoulli Gaussian	CFM	0.066 ± 0.011	0.099 ± 0.003	0.068 ± 0.003	0.041 ± 0.015	0.122 ± 0.030	0.199 ± 0.048
	FBM	0.036 ± 0.008	0.048 ± 0.005	0.037 ± 0.007	0.006 ± 0.007	0.075 ± 0.025	0.115 ± 0.038
Discounted uniform	CFM	0.071 ± 0.006	0.219 ± 0.010	0.113 ± 0.013	0.011 ± 0.009	0.114 ± 0.013	0.201 ± 0.016
	FBM	0.031 ± 0.006	0.102 ± 0.017	0.047 ± 0.012	0.006 ± 0.004	0.050 ± 0.007	0.081 ± 0.012
Hypoexponential	CFM	0.209 ± 0.021	0.321 ± 0.013	0.218 ± 0.035	0.033 ± 0.030	0.446 ± 0.067	0.872 ± 0.162
	FBM	0.109 ± 0.021	0.145 ± 0.021	0.113 ± 0.038	0.030 ± 0.019	0.228 ± 0.062	0.538 ± 0.139
Geometric termination	CFM	0.855 ± 0.040	0.030 ± 0.007	0.108 ± 0.102	0.710 ± 0.093	1.858 ± 0.102	5.003 ± 0.144
	FBM	0.316 ± 0.016	0.025 ± 0.004	0.016 ± 0.006	0.076 ± 0.061	0.854 ± 0.029	1.556 ± 0.195
Markov switching	CFM	0.265 ± 0.045	0.232 ± 0.018	0.174 ± 0.020	0.086 ± 0.089	0.550 ± 0.049	0.493 ± 0.069
	FBM	0.150 ± 0.023	0.051 ± 0.023	0.043 ± 0.021	0.071 ± 0.048	0.375 ± 0.044	0.095 ± 0.026

F.3 Wasserstein Error and Density Resolution

Aggregate accuracy and training behavior. FBM has lower final mean W_1 on all six toys (Table 6). Relative reductions are 5.3%, 45.9%, 56.1%, 47.9%, 63.0%, and 43.4% in table order. These are within-task comparisons; raw distances across different return scales are not directly comparable. Figure 5 shows lower late-training error for FBM on the five non-rare-mode tasks. For the three two-action environments, training curves report the average of the two action-conditional W_1 distances, whereas Table 6 reports the distance between policy-marginal distributions. On the rare-mode chain, FBM reaches a lower error earlier and subsequently deteriorates, leaving only a small final difference. The reported final checkpoints retain this behavior instead of selecting the minimum observed error. These results support improved return-distribution learning under the tested update and integration budgets, without establishing uniform dominance across seeds, quantiles, or density features.

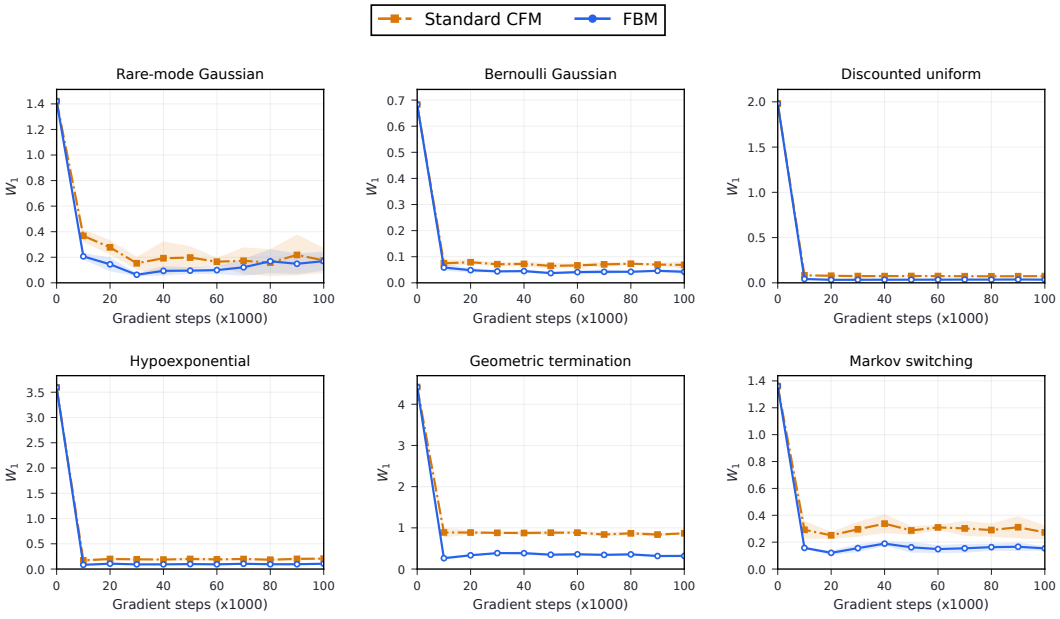

Figure 5: **Training-time distributional error.** Wasserstein-1 error evaluated every 10,000 updates using 4,096 fixed Gaussian inputs and 16 Euler steps. For two-action environments, errors are computed separately for each action and then averaged within each seed. Orange dash-dotted curves denote CFM and blue solid curves denote FBM; bands show one sample standard deviation across five seeds. Each panel uses a linear vertical scale in its own return units.

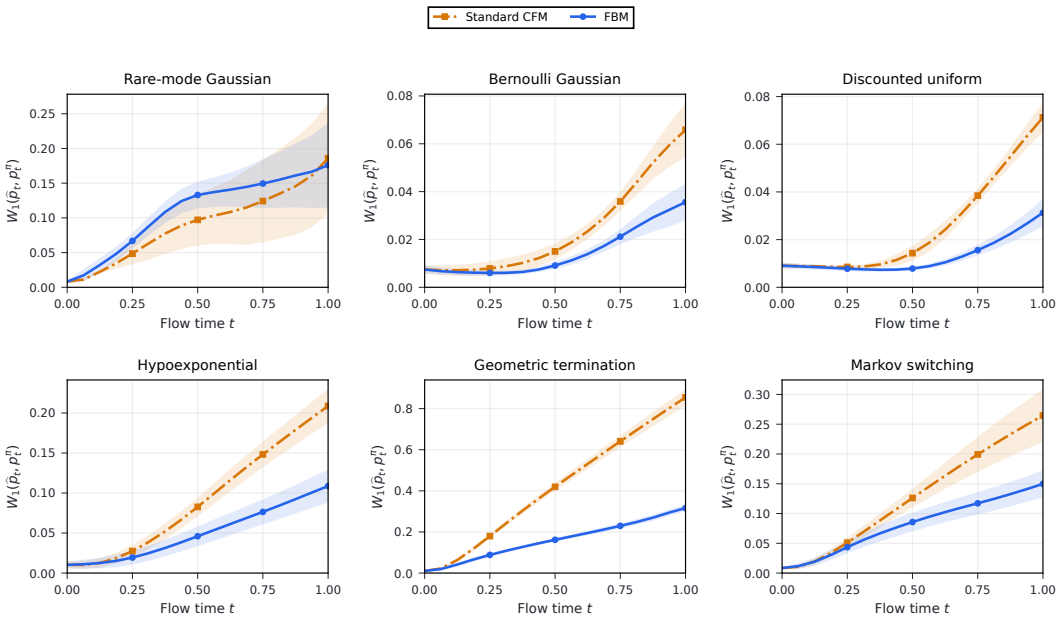

Figure 6: **Distributional error across flow time.** Wasserstein-1 distance between learned flow marginals and independently sampled reference marginals at the final 100,000-update checkpoints. Curves use 16-step Euler integration, with markers at $t = 0.25, 0.5, 0.75, 1$. Two-action environments use 20,000 generated samples per action and seed, combined with equal weight; other environments use 20,000 samples per seed. Reference sample counts are 400,000 for two-action environments, 1,000,000 for rare-mode Gaussian, and 200,000 otherwise. Orange dash-dotted curves denote CFM and blue solid curves denote FBM; bands show one sample standard deviation across five trained seeds. Panels use independent linear vertical scales.

Distributional accuracy across flow time. To evaluate intermediate distributions rather than only terminal returns, we measure $W_1(\widehat{p}_t, p_t^\pi)$ along the learned flow at the final checkpoint (Figure 6). The reference is $p_t^\pi = \mathrm{Law}((1-t)\epsilon + tG^\pi)$, where $\epsilon \sim \mathcal{N}(0,1)$ is independent of the analytic return G^π. For the two-action environments, both learned and reference distributions are marginalized over the initial action under the fixed uniform policy before computing W_1. We evaluate all 17 nodes of a 16-step Euler integration, highlighting $t \in \{0.25, 0.5, 0.75, 1\}$. Distances are computed directly between empirical samples without density smoothing; finite-sample effects therefore remain, including a small nonzero error at $t = 0$.

FBM achieves lower mean W_1 at all four highlighted times on the five non-rare-mode tasks. On the rare-mode chain, its error is higher at the three highlighted intermediate times, despite a slightly lower terminal error. Thus, the gains extend beyond the endpoint on five tasks, but intermediate accuracy is not uniformly improved across environments. These results support improved approximation of the reference flow marginals under the tested settings; isolating the causal contribution of intermediate Bellman supervision requires a matched ablation.

Density estimation and unresolved modes. To distinguish density-estimation artifacts from actual distribution errors, we evaluate the final two-action geometric-termination toy process under different Gaussian KDE bandwidths $h \in \{0.5, 0.05, 0.02\}$ (Figure 7). The same estimator is applied to analytic samples and learned samples for comparison. Large bandwidths smooth out fine-grained modes even for the analytic distribution, while smaller bandwidths reveal that FBM recovers the dominant peaks more accurately than STANDARD-CFM. However, some higher-order modes remain underestimated, indicating that the remaining errors arise from imperfect distribution learning rather than density smoothing alone.

CDF-based interval probabilities provide a check that does not depend on KDE. Averaged across the five seeds, FBM assigns 11.76% of its mass to $[-0.1, 0.1]$ and 8.75% to $[0.9, 1.1]$, compared

with analytic values of 14.32% and 10.87%; CFM assigns 9.90% and 4.47%. FBM still places 1.35% in the inter-mode interval $[1.4, 1.6]$, whose analytic probability is about 0.000118%. The mean 99th percentile is 13.6545 for FBM and 10.2075 for CFM, below the analytic 15.2102. FBM therefore improves major probability masses and tail extent while leaving measurable local mode errors. Separating network capacity, optimization, and numerical integration as causes of those residual errors requires additional controlled comparisons.

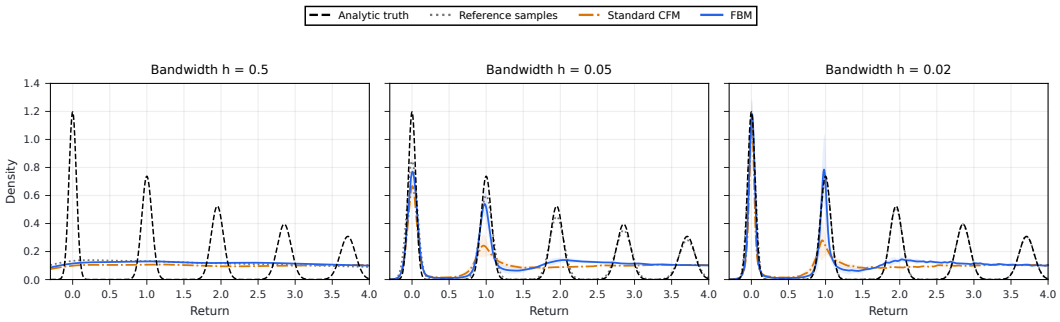

Figure 7: **Density estimation under different KDE bandwidths.** Return densities for the geometric-termination toy process using different Gaussian bandwidths. Black dashed curves denote the analytic density, gray dotted curves denote analytic samples processed by the same estimator, and colored curves show learned distributions. Smaller bandwidths reveal finer modes, while FBM better recovers the dominant probability masses than STANDARD-CFM.

G LIMITATIONS

Although our method improves distributional Bellman learning through flow-time Bellman matching, several limitations remain. First, the proposed framework introduces additional computational overhead compared with conventional scalar or quantile-based critics. Learning and evaluating the time-dependent velocity field requires numerical integration and potentially multiple flow samples, which increases computational cost. Second, while flow-time supervision improves the fidelity of learned return distributions, accurately capturing rare modes in highly multimodal return structures remains challenging. Finally, similar to other offline RL methods, the performance of our approach still depends on the coverage and quality of the offline dataset, and robustness under severe distribution shift remains an important direction for future work.