SOFT-VALUE FLOW MATCHING FOR OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-matching policies represent multimodal actions, but improving them with a learned critic in offline reinforcement learning faces two difficulties: backpropagating through multi-step sampling is costly and unstable, and critic queries on actor-generated actions can expose the policy to out-of-distribution value errors. We introduce Soft-Value Flow Matching (SVFM), which trains a multi-step actor with the gradient of a conditional soft value for KL-regularized policy improvement. Starting from noised data actions, SVFM samples stochastic continuations of a fixed behavior process and evaluates the critic at their endpoints, keeping the queries used for guidance independent of the current actor. A log-mean-exp of these values provides a Monte Carlo target for a soft-value network, whose gradient guides actor updates without trajectory backpropagation. We further propose an adaptive temperature that scales with the spread of critic values across continuations. SVFM achieves higher average success rates than the compared baselines on OGBench, LIBERO, and LIBERO-Plus. It also improves real-world manipulation with both task-specific flow policies and a pretrained VLA model.

1 INTRODUCTION

Flow-matching and diffusion policies are widely used in continuous control due to their ability to represent complex, multimodal action distributions (Chi et al., 2023), for example as action experts in vision-language-action (VLA) models (Shukor et al., 2025; Reuss et al., 2025; Cheang et al., 2025; Zheng et al., 2025). Recent work also combines them with offline reinforcement learning (RL), using a learned critic to improve beyond behavior cloning (BC) (Wang et al., 2023; Hansen-Estruch et al., 2023; Park et al., 2025). Because these policies generate actions through multiple denoising or integration steps, a central challenge is to improve them with the critic without backpropagation through time (BPTT) across the entire sampling trajectory (Li & Levine, 2026).

Prior work avoids multi-step BPTT in two ways. *(i) Bypassing multi-step actor updates.* Test-time selection or guidance keeps a pretrained actor fixed but adds inference-time computation (Hansen-Estruch et al., 2023; Lu et al., 2023; Li et al., 2024; Ruan et al., 2026), while one-step distillation enables direct critic maximization at the cost of expressivity (Park et al., 2025). *(ii) Training multi-step actors with intermediate guidance.* One approach learns intermediate values from critic values at actor endpoints (Doo et al., 2026). Others inject critic gradients at intermediate actions into the policy's score or velocity targets (Psenka et al., 2023; Fang et al., 2025; Alles et al., 2025). Stochastic optimal control (SOC) frames KL-regularized policy improvement (Dai Pra, 1991; Kappen, 2005; Fleming & Soner, 2006) as balancing critic value against deviation from the behavior policy π_β (Peters et al., 2010; Peng et al., 2019; Nair et al., 2020); within it, adjoint-based methods propagate terminal critic gradients backward along actor trajectories (Li & Levine, 2026).

However, avoiding BPTT does not by itself ensure reliable critic-based supervision in offline RL. A critic learned from a fixed dataset can overestimate out-of-distribution (OOD) actions, leading the actor to exploit these errors rather than improve returns (Fujimoto et al., 2019; Kumar et al., 2019; Levine et al., 2020). When guidance is constructed from actor samples, actor updates also shift where the critic is evaluated, potentially toward overestimated actions. In a toy experiment (Figure 1), actor-supervision queries on actor samples extend into an overestimated OOD region, whereas those from fixed behavior continuations remain near the data.

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.

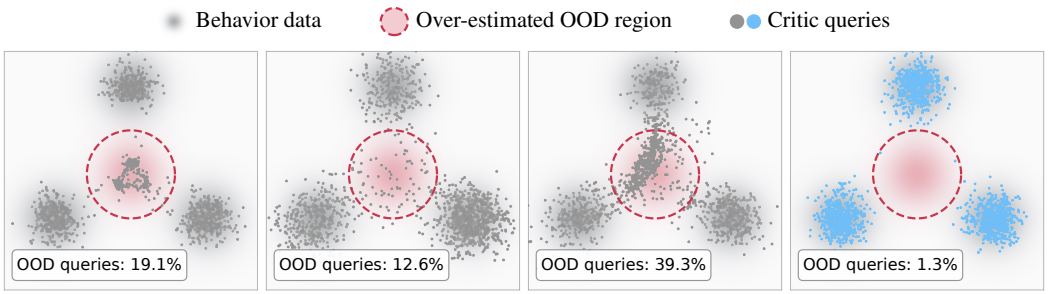


Figure 1: Critic queries for actor supervision when the critic over-estimates an OOD region in a toy task. Queries on actor samples enter that region, whereas our method evaluates the critic there much less often. Experimental details are in Appendix I.1; the corresponding analysis on the OGBench benchmark (Park et al., 2024) is in Appendix I.2.

Our key insight is to learn a function of intermediate actions that aggregates terminal critic values over behavior continuations, and use its gradient to supervise the actor. The Doob h-transform identifies this function as a conditional *soft value*, whose scaled gradient gives the optimal SOC correction (Doob, 1957; Chen et al., 2021); it places greater weight on higher-value outcomes (Uehara et al., 2024; 2025). Motivated by this insight, we introduce Soft-Value Flow Matching (SVFM): starting from noisy actions constructed from data, SVFM samples stochastic continuations of a fixed behavior process and evaluates the critic at their endpoints. These evaluations provide Monte Carlo regression targets for a soft-value network, whose input gradient supervises the flow actor through a guided regression objective (Psenka et al., 2023; Alles et al., 2025; Doo et al., 2026). This soft value depends on a temperature that controls how strongly it favors higher-value continuations; with a constant temperature, this preference strengthens as critic values spread out and weakens as they concentrate. We therefore introduce an adaptive temperature that changes during training with the spread of critic values across behavior continuations, keeping this preference consistent.

Our contributions are threefold:

- We propose Soft-Value Flow Matching (SVFM), a method for KL-regularized policy improvement that trains a multi-step flow actor with soft-value guidance learned from behavior continuations, without trajectory BPTT.
- We introduce an adaptive temperature that changes during training with the spread of critic values across behavior continuations.
- SVFM achieves the highest average score among the compared methods on OGBench (Park et al., 2024), as well as on LIBERO (Liu et al., 2023a) and LIBERO-Plus (Fei et al., 2025) when fine-tuning a pretrained VLA model. It further improves real-world robot policies.

2 Preliminaries

Offline reinforcement learning. We consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ with continuous action space $\mathcal{A} \subset \mathbb{R}^d$ and a fixed dataset $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$ collected by a behavior policy π_β (Sutton & Barto, 2018; Levine et al., 2020). We denote by $\pi(a \mid s)$ the policy to be optimized and by $Q(s, a)$ its learned action-value function. Because policy improvement can exploit inaccurate values for out-of-distribution actions, offline RL commonly constrains the learned policy toward data-supported actions through behavior regularization or conservative value estimation (Levine et al., 2020; Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2019; Fujimoto & Gu, 2021; Kumar et al., 2020). For a temperature parameter $\lambda > 0$, KL-regularized policy improvement balances high predicted return against deviation from the behavior policy (Wu et al., 2019; Nair et al., 2020). For each state s, it solves $\pi^\star = \arg\max_\pi \big[\mathbb{E}_{a\sim\pi(\cdot|s)}[Q(s,a)] - \lambda D_{\mathrm{KL}}(\pi(\cdot \mid s) \,\|\, \pi_\beta(\cdot \mid s))\big]$. Its statewise optimum has the closed form (Peng et al., 2019; Nair et al., 2020)

$$\pi^\star(a \mid s) \propto \pi_\beta(a \mid s) \exp\big(Q(s,a)/\lambda\big). \tag{1}$$

Flow policies. A flow policy transports Gaussian noise to actions through a time-dependent velocity field. We use generation time $t \in [0,1]$ and the linear interpolant $x_t = (1-t)x_0 + ta$, where $x_0 \sim \mathcal{N}(0, I)$ and $a \sim \pi_\beta(\cdot \mid s)$ are independent. This defines a family of marginal distributions p_t, where p_t denotes the distribution of x_t at time t, with $p_0 = \mathcal{N}(0, I)$ and $p_1 = \pi_\beta$. Conditional flow matching learns a time-dependent velocity field u_ϕ via (Lipman et al., 2023; Liu et al., 2023b)

$$\mathcal{L}_{\mathrm{CFM}}(\phi) = \mathbb{E}_{t,x_0,a}\big\|u_\phi(x_t,t) - (a - x_0)\big\|^2, \tag{2}$$

where $t \sim \mathcal{U}[0,1]$. Its population minimizer is the behavior field $u(x,t) = \mathbb{E}[a - x_0 \mid x_t = x]$, with u_ϕ its learned approximation. The ODE $\dot{x}_t = u(x_t,t)$ has marginals p_t and therefore transports Gaussian noise to π_β. For the linear interpolant, the marginal score is determined by the behavior field, so no separate score network is needed (Albergo et al., 2023):

$$\nabla_x \log p_t(x) = -\frac{x - t\,u(x,t)}{1-t}, \qquad 0 < t < 1. \tag{3}$$

For any diffusion schedule $\sigma_t > 0$, the reverse SDE with drift $b(x,t) = u(x,t) + \frac{\sigma_t^2}{2}\nabla_x \log p_t(x)$ and diffusion coefficient σ_t has the same marginals p_t as the ODE under standard regularity conditions (Song et al., 2021). Following the memoryless schedule of adjoint matching, we set $\sigma_t^2 = 2(1-t)/t$ (Domingo-Enrich et al., 2025; Li & Levine, 2026). Substituting equation 3 gives

$$dX_t = \Big(2\,u(X_t,t) - \frac{X_t}{t}\Big)dt + \sqrt{\frac{2(1-t)}{t}}\,dW_t. \tag{4}$$

The dynamics are defined for $0 < t < 1$, with endpoint laws understood as limits. Here X_t denotes the stochastic trajectory of equation 4; unlike the interpolant $x_t = (1-t)x_0 + ta$, it follows a different process while sharing the same marginal distribution p_t at each time. Under this process, X_0 and X_1 are independent conditional on s (Domingo-Enrich et al., 2025).

Policy improvement as stochastic optimal control. KL-regularized policy improvement for diffusion and flow policies can be formulated as a stochastic optimal control problem (Uehara et al., 2024; Tang, 2024; Domingo-Enrich et al., 2025). With b denoting the base drift above and c an additional control, QAM (Li & Levine, 2026) applies this formulation to offline RL, using Q as the terminal reward and the memoryless behavior process as the base dynamics:

$$\begin{aligned} \max_{c} \quad & \mathbb{E}_c\Big[Q(s,X_1) - \lambda \int_0^1 \frac{\|c(X_t,t)\|^2}{2\sigma_t^2}\,dt\Big] \\ \text{subject to} \quad & dX_t = [b(X_t,t) + c(X_t,t)]dt + \sigma_t\,dW_t. \end{aligned} \tag{5}$$

Let P_c and P_β be the path laws of the controlled and base processes with the same initial law and diffusion coefficient. Under the conditions of Girsanov's theorem (Girsanov, 1960),

$$\mathbb{E}_c\left[\int_0^1 \frac{\|c(X_t,t)\|^2}{2\sigma_t^2}\,dt\right] = D_{\mathrm{KL}}(P_c \,\|\, P_\beta). \tag{6}$$

Thus, equation 5 trades terminal value against trajectory-level deviation from the behavior process (Domingo-Enrich et al., 2025; Uehara et al., 2024). For the memoryless process equation 4, $X_0 \perp X_1$ ensures that terminal reweighting by $\exp(Q(s,X_1)/\lambda)$ preserves the initial noise law, and the optimum realizes equation 1; the trajectory KL reduces to the terminal-policy KL at the optimum, which is not guaranteed for general noise schedules (Domingo-Enrich et al., 2025).

QAM solves equation 5 by regressing the control onto detached targets obtained from a lean adjoint propagated along controlled trajectories (Domingo-Enrich et al., 2025; Li & Levine, 2026). This avoids backpropagation through time through the learned sampler, while target construction still uses the terminal critic gradient and a backward recursion through Jacobians of the base drift.

3 Method: Soft-Value Flow Matching

We train a multi-step flow policy using soft-value guidance estimated from continuations of a fixed behavior model. To this end, we start from noisy data actions, sample stochastic continuations of a fixed behavior process, and query the critic at their terminal actions. We aggregate these values into a soft-value regression target for intermediate actions and use the learned soft value's input gradient to guide the actor. Under exact components, established h-transform, probability-flow, and flow-matching identities show that this guidance recovers the KL-regularized target (Chen et al., 2021; Song et al., 2021; Lipman et al., 2023).

The h-transform and the soft value. For a fixed environment state s, P_β is the path law of the memoryless reference reverse SDE in equation 4, which shares the marginals p_t of the behavior ODE; $\mathbb{E}_\beta$ denotes expectation under this law. The optimal control $c^\star$ is the additional drift that maximizes equation 5: the controlled SDE has drift $b+c^\star$ and the same diffusion coefficient σ_t as the reference process. For the idealized analysis, we fix Q and $\lambda > 0$ and assume $|Q(s,a)| \le M_Q < \infty$ on the relevant endpoint supports. This ensures finite positive exponential expectations (Appendix A.3) but is not guaranteed for an unconstrained neural critic. Under these and the regularity and endpoint-limit conditions (Appendix B), the Doob h-transform defines (Doob, 1957; Chen et al., 2021)

$$h(x,t) = \mathbb{E}_\beta\big[e^{Q(s,X_1)/\lambda} \mid X_t = x\big].$$

The expectation is over terminal actions of the behavior process continued from (x,t), with s held fixed. Defining the soft value (Uehara et al., 2025; Li et al., 2024), the optimal additional drift becomes its gradient scaled by the noise level and temperature:

$$V_\lambda(s,x,t) := \lambda \log h(x,t), \qquad c^\star(x,t) = \frac{\sigma_t^2}{\lambda}\,\nabla_x V_\lambda(s,x,t). \tag{7}$$

Learning the soft-value network. We train the critic Q_ω on clean actions by standard off-policy TD regression on transitions from the offline dataset $\mathcal{D}$:

$$\mathcal{L}_Q(\omega) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D},\ a'\sim\pi_\theta(\cdot|s')}\Big[\big(Q_\omega(s,a) - \big(r + \gamma Q_{\bar\omega}(s',a')\big)\big)^2\Big].$$

Here a' is sampled from the current flow actor and $\bar\omega$ denotes the target parameters. We also use $Q_{\bar\omega}$ for Q in the soft-value regression below. At a data-path anchor $x_t = (1-t)x_0 + ta$, we reintegrate the learned behavior field u_ϕ under equation 4 with K independent Brownian paths to obtain terminal actions $\{a^{(j)}\}_{j=1}^K$. We define the Monte Carlo soft-value target and fit a soft-value network V_ψ to it:

$$\widehat{V}_K(s,x_t,t) := \lambda \log\Big(\frac{1}{K}\sum\nolimits_{j=1}^{K} e^{Q(s,a^{(j)})/\lambda}\Big),$$
$$\mathcal{L}_V(\psi) = \mathbb{E}_{t,x_0,a}\Big[\big(V_\psi(s,x_t,t) - \mathrm{sg}[\widehat{V}_K(s,x_t,t)]\big)^2\Big].$$

By Jensen's inequality (Jensen, 1906), the finite-K target underestimates equation 7 in expectation, with bias vanishing as K grows under the bounded-critic assumption; Appendix A.3 gives the $1/K$ leading bias, an $O(K^{-3/2})$ bound, and a sharper $O(K^{-2})$ remainder.

Policy improvement via the probability-flow ODE. To train the flow actor toward $\pi^\star$ in equation 1, we convert the SDE controlled by the h-transform (Uehara et al., 2024) into an ODE with the same time marginals, using the standard probability-flow construction (Song et al., 2021). The optimal path law reweights the behavior path law by $\exp(Q(s,X_1)/\lambda)$; taking the conditional expectation of this terminal weight given $X_t = x$ gives $h(x,t)$, so the optimally controlled marginal $p_t^\star$ satisfies $p_t^\star(x) \propto p_t(x)h(x,t)$. Consequently, $\nabla_x \log p_t^\star = \nabla_x \log p_t + \frac{1}{\lambda}\nabla_x V_\lambda$. The standard probability-flow conversion gives the following velocity; Appendix A.1 derives it in our notation:

$$v^\star(x,t) = u(x,t) + \frac{\sigma_t^2}{2\lambda}\nabla_x V_\lambda(s,x,t) = u(x,t) + \frac{1-t}{t\lambda}\nabla_x V_\lambda(s,x,t). \tag{8}$$

Thus, $(1-t)/(t\lambda)$ is set by the memoryless schedule, not chosen as a guidance schedule. We instantiate a sample-level guided regression objective (Psenka et al., 2023; Alles et al., 2025; Doo et al.,

Proposition 1 (Proof in Appendix A.1). *Fix an environment state s, a bounded critic Q, a temperature $\lambda > 0$, and $\kappa \in (0,1]$, with λ and κ constant over flow time. Let u be the exact CFM field associated with π_β, and let $V_{\lambda,\kappa}$ be the exact soft value under the scaled behavior process equation 10 with $u_\phi = u$. Assume the regularity and endpoint-limit conditions of Appendix B for this process. Then, with the endpoints understood in the boundary-limit sense, integrating*

$$\dot{x}_t = u(x_t,t) + \kappa^2\frac{1-t}{t\lambda}\,\nabla_x V_{\lambda,\kappa}(s,x_t,t)$$

from $x_0 \sim \mathcal{N}(0,I)$ to $t=1$ produces samples from $\pi^\star(a \mid s) \propto \pi_\beta(a \mid s)\exp(Q(s,a)/\lambda)$.

2026) with our continuation-trained soft value and $\beta_t = (1-t)/(t\lambda)$, regressing the actor velocity onto the data velocity plus the scaled soft-value gradient:

$$\mathcal{L}_{\text{actor}}(\theta) = \mathbb{E}_{t,x_0,a}\Big\| v_\theta(x_t,t) - \text{sg}\big[(a-x_0) + \beta_t\, \nabla_x V(s,x_t,t)\big]\Big\|^2. \tag{9}$$

We use $V = V_\psi$. This objective follows the same regression principle as equation 2. The sample-level target $(a - x_0)$ has conditional mean $u(x,t)$, while $\nabla_x V$ is fixed given x_t, so the value correction passes unchanged through the conditional expectation. Thus, equation 9 learns equation 8 without constructing its velocity target explicitly, just as CFM learns u from sample-level targets. When $V_\psi = V_\lambda$, the pointwise population minimizer of equation 9 is exactly equation 8; the derivation is given in Appendix B.1.

Proposition 1 is a fixed-objective population result for exact components, not a convergence guarantee for adaptive-temperature training. The unconditioned $V_\psi(s,x,t)$ tracks changing targets as Q and the update-wise temperature change; Appendices A.1 and E state this limitation explicitly.

Controlling continuation stochasticity with κ. Under the original memoryless schedule, the distribution of terminal actions becomes independent of the starting anchor as $t \to 0$, and guidance approaches a common mean shift, as shown in Appendix C. This limits anchor-specific guidance: the same correction is applied even to anchors whose behavior-ODE continuations would reach different actions. To control anchor dependence at finite $t > 0$, we introduce $\kappa \in (0,1]$, scaling the diffusion coefficient by κ and the score correction by κ^2. Under suitable regularity, the continuation approaches the deterministic behavior ODE as $\kappa \to 0$ with $t > 0$ fixed, a limit distinct from taking $t \to 0$. The guidance coefficient and continuation SDE become

$$\beta_t^{(\kappa)} = \kappa^2 \frac{1-t}{t\lambda}, \quad dx_t = \left[u_\phi(x_t,t) - \kappa^2 \frac{x_t - t\,u_\phi(x_t,t)}{t}\right] dt + \kappa\sqrt{\frac{2(1-t)}{t}}\, dW_t. \tag{10}$$

Let P_κ denote the path law induced by this scaled behavior process and $V_{\lambda,\kappa}$ the soft value equation 7 with expectation under P_κ. We train V_ψ on these continuations and use $\beta_t^{(\kappa)}$ in equation 9; setting $\kappa = 1$ recovers the original schedule. The soft value and guidance coefficient must use the same κ, as in Proposition 1.

Adaptive temperature. With a constant temperature λ, larger differences in Q give high-value actions more relative weight in the soft-value target, so the same temperature is not comparable across critic scales. TD3+BC (Fujimoto & Gu, 2021) normalizes the actor's Q term by the mean absolute Q value to calibrate its balance with behavior cloning. We apply this normalization principle to the temperature, setting λ relative to the variation of Q across terminal actions sampled from each anchor. Let $s_Q := \mathbb{E}_{x_t}[\text{std}_{j=1,\dots,K}\, Q(s,a^{(j)})]$ denote the within-anchor standard deviation of Q, averaged over anchors. For a dimensionless gain $g > 0$, we set

$$\lambda = \frac{\kappa^2 s_Q}{g}, \qquad \text{so that} \qquad \beta_t^{(\kappa)} = \frac{\kappa^2(1-t)}{t\lambda} = \frac{g\,(1-t)}{t\, s_Q}. \tag{11}$$

The κ^2 factors cancel, leaving an actor coefficient proportional to g and inversely proportional to s_Q. Guidance still depends on κ through the terminal-action distribution, s_Q, and the soft value. At optimization update n, we estimate s_Q, detach the resulting scalar λ_n, and hold it fixed across flow time in that update's soft-value target and actor coefficient as described in Appendix E. It may change between updates. For a fixed continuation law, positive rescalings of Q also rescale λ, preserving exact guidance as established in Proposition 2. Further details are in Appendix D.

Proposition 2 (Proof in Appendix A.2)**.** *Fix the continuation law, anchor distribution, $\kappa \in (0,1]$, and $g > 0$, and assume $0 < s_Q < \infty$. Let $Q'(s,a) = cQ(s,a) + b(s)$ with $c > 0$, and recompute the adaptive temperature using Q' and the same sampling scheme. Let V and V' denote the corresponding finite, differentiable exact soft values. Then $s_{Q'} = c\, s_Q$, $\lambda' = c\lambda$, and $V' = cV + b(s)$, so*

$$\beta_t^{(\kappa)\prime}\, \nabla_x V' = \beta_t^{(\kappa)}\, \nabla_x V.$$

Thus the exact guidance correction, and hence the actor regression target, is invariant to this positive affine transformation of Q.

Table 1: OGBench success rates (%) under the standard setting (Park et al., 2025), averaged over the 5 tasks of each domain, 8 seeds. Bold marks the best result per row. Full results are in Table 12.

	Gaussian		Diffusion		Flow					
Environment	IQL	ReBRAC	IDQL	CAC	FBRAC	IFQL	FQL	CEP	Q-Flow	SVFM (ours)
antmaze-large	53 ±3	81 ±5	21 ±5	33 ±4	60 ±6	28 ±5	79 ±3	63 ±5	87 ±5	**90** ±2
antmaze-giant	4 ±1	26 ±8	0 ±0	0 ±0	4 ±4	3 ±2	9 ±6	8 ±2	**32** ±7	21 ±9
humanoidmaze-medium	33 ±2	22 ±8	1 ±0	53 ±8	38 ±5	60 ±14	58 ±5	59 ±3	**83** ±4	82 ±4
humanoidmaze-large	2 ±1	2 ±1	1 ±0	0 ±0	2 ±0	11 ±2	4 ±2	3 ±1	7 ±2	**20** ±5
antsoccer	8 ±2	0 ±0	12 ±4	2 ±4	16 ±1	33 ±6	60 ±2	**65** ±3	55 ±3	58 ±3
scene	28 ±1	41 ±3	46 ±3	40 ±7	45 ±5	30 ±3	56 ±2	35 ±8	**61** ±1	60 ±1
puzzle-3x3	9 ±1	21 ±1	10 ±2	19 ±0	14 ±4	19 ±1	30 ±1	25 ±3	48 ±4	**68** ±3
puzzle-4x4	7 ±1	14 ±1	29 ±3	15 ±3	13 ±1	25 ±5	17 ±2	5 ±2	29 ±2	**33** ±4
cube-single	83 ±3	91 ±2	95 ±2	85 ±9	79 ±7	79 ±2	**96** ±1	**96** ±1	94 ±2	94 ±1
cube-double	7 ±1	12 ±1	15 ±6	6 ±2	15 ±3	14 ±3	29 ±2	**57** ±5	32 ±5	42 ±3
Average	23 ±1	31 ±1	23 ±1	25 ±2	29 ±1	30 ±2	44 ±1	42 ±1	53 ±1	**57** ±1

Table 2: OGBench success rates (%) under the advanced setting (Li & Levine, 2026), averaged over the 5 tasks of each domain, 12 seeds. Bold marks the best result per row. "-sparse" denotes sparse-reward variants. Full results are in Table 13.

	Gaussian	Diffusion		Flow						
Environment	ReBRAC	DAC	QSM	FBRAC	IFQL	FQL	QAM	QAM-E	Q-Flow	SVFM (ours)
antmaze-large	**94** ±1	88 ±4	90 ±4	2 ±3	36 ±6	76 ±6	80 ±5	79 ±4	93 ±0	90 ±1
antmaze-giant	**57** ±6	16 ±8	24 ±9	0 ±0	1 ±2	0 ±0	15 ±5	0 ±0	41 ±6	48 ±5
humanoidmaze-medium	69 ±7	83 ±4	82 ±4	39 ±4	**86** ±2	68 ±9	62 ±7	51 ±16	84 ±3	83 ±7
humanoidmaze-large	17 ±4	0 ±0	6 ±2	0 ±0	24 ±5	9 ±4	7 ±3	2 ±1	6 ±2	**25** ±3
scene-sparse	65 ±7	68 ±4	78 ±3	50 ±12	84 ±2	78 ±3	97 ±1	97 ±1	97 ±1	**98** ±2
puzzle-3x3-sparse	79 ±10	68 ±11	57 ±10	0 ±1	**100** ±0	70 ±16	**100** ±0	**100** ±0	**100** ±0	**100** ±0
puzzle-4x4-sparse	0 ±0	0 ±0	0 ±0	15 ±7	0 ±0	5 ±4	0 ±0	**39** ±8	0 ±0	0 ±0
cube-double	9 ±2	35 ±3	33 ±3	0 ±0	11 ±2	46 ±4	63 ±5	62 ±3	36 ±3	**69** ±2
cube-triple	1 ±1	5 ±3	**6** ±1	0 ±1	0 ±0	3 ±2	3 ±1	5 ±1	3 ±1	3 ±1
cube-quadruple	9 ±4	3 ±4	19 ±2	0 ±0	2 ±2	2 ±4	2 ±1	5 ±3	10 ±6	**50** ±2
Average	40 ±2	37 ±2	40 ±2	11 ±2	34 ±1	36 ±2	43 ±1	44 ±2	47 ±1	**56** ±1

4 EXPERIMENTS

OGBench. We evaluate on OGBench (Park et al., 2024), a long-horizon offline RL benchmark spanning locomotion and manipulation, under the *standard* (Park et al., 2025) and *advanced* (Li & Levine, 2026) settings. The latter adds action chunking for manipulation, expands the critic ensemble from 2 to 10, and uses pessimistic value learning. Both settings report domain-level success rates averaged over five tasks. All flow policies use four hidden layers of width 512 (Appendix F). Following the baselines' per-environment tuning protocol, we tune only SVFM's κ and g, keeping all other settings fixed (Appendix G). We also study offline-to-online fine-tuning and adaptive temperature. Appendix M reports additional experiments and analyses, including a FAC-protocol comparison, a behavior-versus-actor continuation ablation, and wall-clock training and inference costs.

LIBERO and LIBERO-Plus. LIBERO (Liu et al., 2023a) evaluates language-conditioned robotic manipulation, while LIBERO-Plus (Fei et al., 2025) tests robustness under task perturbations. We fine-tune $\pi_{0.5}$ (Black et al., 2025) using 5 demonstrations per task, starting from a BC policy trained on the same data. Further experimental details are provided in Appendix N.

Real-world. Our real-robot study covers 4 manipulation tasks, each with 50 teleoperated demonstrations. We consider task-specific flow policies and a shared pretrained VLA policy, GR00T N1.7 (Bjorck et al., 2025), fine-tuned by behavior cloning for 10K (BC) or 20K (BC+) updates; SVFM starts from BC. Beyond success rates, we compare successful trajectories and report path length, gripper switches, and joint-velocity jitter (Appendix J.2).

Our experiments address the following research questions:

RQ1. Does SVFM outperform existing offline RL methods on OGBench? (Section 4.1)

RQ2. Can our method improve further through online fine-tuning? (Section 4.2)

RQ3. Does adaptive temperature outperform constant temperature? (Section 4.3)

RQ4. Can our approach be extended to VLA policies and real-world tasks? (Sections 4.4 and 4.5)

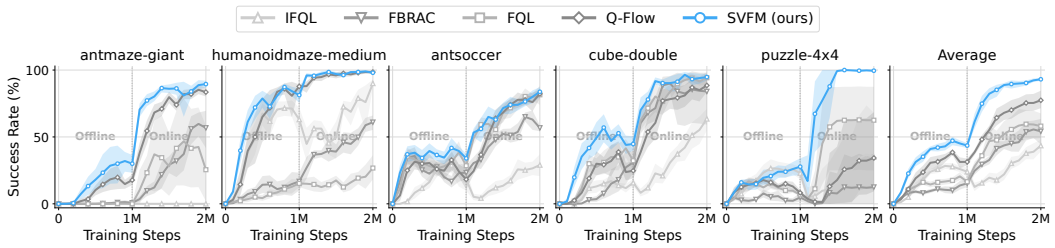


Figure 2: Offline-to-online results on 5 OGBench domains under the protocol of Doo et al. (2026), using each domain's default task: 1M offline gradient updates, followed by 1M online environment steps. 8 seeds. Experimental details are provided in Appendix K.

4.1 Offline RL Results

Tables 1 and 2 show that SVFM leads all baselines on average: 57% versus Q-Flow's 53% in the standard setting and 56% versus 47% in the advanced setting. SVFM reaches 68% on puzzle-3x3 versus 48% for Q-Flow in the standard setting and 50% on cube-quadruple versus 19% for QSM in the advanced setting. One possible contributor is that SVFM learns soft-value guidance from fixed-behavior continuations, so actor updates do not shift the critic queries used for policy supervision toward potentially overestimated actions. Appendix I.2 shows that SVFM's actor-supervision queries are less often valued above all sampled behavior actions than those of the baselines. This mechanism concerns how critic values guide the actor rather than how accurate the critic is; we examine whether SVFM complements advanced critic learning in Appendix M.1.

4.2 Offline-to-Online RL Results

Figure 2 compares offline-to-online performance across the 5 domains, where each method is first trained offline and then fine-tuned with online interaction. During online fine-tuning, new transitions enter the replay buffer, and SVFM uses either the current policy or a periodically refreshed copy as its continuation source, selected per domain (Appendix G.3). With online interaction, our approach continues to improve beyond its offline performance and achieves the highest final average success rate across the 5 domains, with particularly pronounced gains on puzzle-4x4. These results demonstrate that the method is not limited to the frozen-reference setting of offline training and can leverage online interaction for further policy improvement. Implementation details for online adaptation, including the online protocol and evaluation schedule, are provided in Appendix K.

4.3 Adaptive Temperature vs. Constant Temperature

In Table 3, we compare adaptive temperature with 4 constant temperatures, $\lambda = 0.1, 0.5, 1.0$, and 3.0, under the standard setting. All runs share κ; adaptive temperature uses the per-domain g from Table 1. Adaptive temperature achieves 56% average success, exceeding both the best single constant temperature (45%) and the best constant chosen separately for each domain in hindsight (51%). Although not uniformly best, it ranks first on 6 of 10 domains. With a fixed temperature, changes in the spread of Q-values during training alter the relative emphasis that the soft value places on higher-value continuations. We hypothesize that scaling λ with the evolving spread of terminal Q-values helps accommodate these changes in critic scale during training, keeping this emphasis more consistent rather than relying on a single fixed absolute temperature that does not respond to how the critic scale evolves.

Table 3: Comparison of adaptive and constant λ: success rates (%) at 1M steps on default tasks (3 seeds), with κ from Table 1. *Best*: best constant λ per domain.

		Constant λ				
Environment	Adaptive λ	0.1	0.5	1.0	3.0	*Best*
antmaze-large	**93**±2	91±4	81±8	67±8	12±6	91±4
antmaze-giant	0±0	**1**±1	0±0	0±0	0±0	**1**±1
humanoidmaze-medium	**94**±2	6±8	67±9	3±1	2±2	67±9
humanoidmaze-large	**26**±9	0±0	9±3	1±1	0±0	9±3
antsoccer	**33**±6	15±6	31±2	19±7	5±2	31±2
scene	91±9	57±29	**100**±0	93±8	71±18	**100**±0
cube-single	95±3	81±6	**98**±3	97±2	95±1	**98**±3
cube-double	**44**±19	4±4	25±11	16±7	1±1	25±11
puzzle-3x3	**65**±15	15±16	26±13	61±15	10±3	61±15
puzzle-4x4	21±6	1±1	9±2	20±3	**26**±3	**26**±3
Average	**56**±3	27±4	45±2	38±2	22±2	51±2

4.4 Language-Conditioned Manipulation: LIBERO and LIBERO-Plus

Table 4 compares SVFM with alternatives for the pretrained VLA policy $\pi_{0.5}$ on LIBERO and LIBERO-Plus. All methods start from behavior cloning (BC) with 5 demonstrations per task. Filtered BC and SVFM add 5K actor updates using demonstrations and successful policy rollouts. SVFM trains a critic on the demonstrations and 2,000 policy rollouts, including failures, for 7K updates (2K before and 5K during actor training). Best-of-N (BoN) and QGF (Zhou et al., 2026), a test-time gradient guidance method for flow policies, use a critic trained on the same data for 7K updates, matching SVFM's critic update count; at test time, BoN selects among 8 BC action chunks, while QGF adds value guidance during sampling. We also compare with Q-VGM (Wang et al., 2026), which converts critic action gradients into velocity-matching targets without trajectory BPTT.

Table 4: Success rates (%) of $\pi_{0.5}$ (Black et al., 2025) on LIBERO and LIBERO-Plus. Bold marks the best result. Experimental details are provided in Appendix N.

	BC	Filtered BC	QGF	BoN	Q-VGM	SVFM (ours)
LIBERO (by task suite)						
Spatial	92 ±1	**96** ±1	91 ±0	92 ±0	92 ±2	95 ±1
Object	78 ±1	81 ±1	**95** ±2	81 ±2	86 ±1	90 ±2
Goal	85 ±0	90 ±1	83 ±1	88 ±2	**93** ±1	89 ±1
Long	52 ±0	60 ±2	52 ±2	46 ±2	47 ±1	**71** ±2
Average	77 ±0	82 ±0	80 ±1	77 ±0	80 ±1	**86** ±1
LIBERO-Plus (by perturbation)						
Camera	52 ±1	55 ±0	50 ±0	49 ±1	52 ±1	**61** ±0
Robot	62 ±1	64 ±1	59 ±0	59 ±0	60 ±1	**66** ±1
Language	69 ±0	71 ±1	69 ±1	67 ±2	70 ±1	**75** ±1
Light	79 ±2	79 ±1	81 ±1	78 ±1	82 ±1	**86** ±1
Background	74 ±1	79 ±1	76 ±2	76 ±1	77 ±1	**83** ±1
Noise	63 ±1	**73** ±0	57 ±0	59 ±0	66 ±1	**73** ±1
Layout	69 ±0	71 ±1	71 ±1	69 ±0	72 ±1	**75** ±0
Average	66 ±0	69 ±0	65 ±0	64 ±0	67 ±0	**73** ±1

We evaluate 50 episodes per LIBERO task and 10,030 perturbed tasks on LIBERO-Plus. In Table 4, our method achieves an average success rate of 86% on LIBERO, compared with 77% for BC, 82% for Filtered BC, 77% for BoN, 80% for QGF, and 80% for Q-VGM. The largest improvement over BC is on LIBERO-Long (52% to 71%), followed by LIBERO-Object (78% to 90%). On LIBERO-Plus, our method achieves 73%, compared with 66% for BC and at most 69% for the other methods, and it is the best or tied for the best under every perturbation type. These results support using our approach to fine-tune a pretrained VLA policy.

4.5 Real-World Manipulation

We compare our method with baselines on 4 real-world manipulation tasks: carrot in pot, bowl stacking, triple bowl stacking, and cube stacking, as illustrated in Figure 6. Each task uses 50 demonstrations, and each method is evaluated over 20 trials per task.

Flow matching policies. Figure 3 compares Flow BC, BoN (IQL critic), Q-Flow, and SVFM using task-specific flow policies with ResNet-18 (He et al., 2016) visual encoders and an MLP-based flow-matching actor. Our method achieves 64% average success, versus 15% for Flow BC, 31% for BoN, and 54% for Q-Flow, and matches or exceeds the baselines on every task. On triple bowl stacking, neither Flow BC nor BoN completes the full sequence, whereas Q-Flow and SVFM reach 30%. On cube stacking, Flow BC fails in all 20 trials while SVFM reaches 50%.

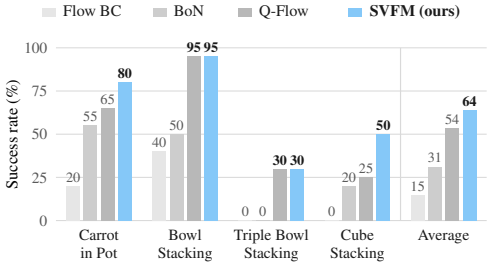


Figure 3: Real-world success rates (%) of task-specific flow policies over 20 trials per task.

VLA. We also fine-tune GR00T N1.7 (Bjorck et al., 2025) jointly on the same demonstrations from all 4 tasks (Table 5). SVFM trains for 10K updates from GR00T behavior-cloned for 10K updates, while BC+ uses 20K behavior-cloning updates, matching SVFM's total actor updates. SVFM reaches 99% average success versus 73% for BC+ and 61% for BoN. The gap is largest on cube stacking, suggesting that critic guidance helps with placement even for a pretrained VLA.

Table 5: GR00T N1.7 real-world success rate (%, 20 trials per task) and median end-effector path length (m) over successful trials. BoN and the BC checkpoint SVFM starts from are reported in Appendix J (Table 16).

	BC+		BoN		SVFM (ours)	
Task	SR (%)	Path (m)	SR (%)	Path (m)	SR (%)	Path (m)
Carrot in pot	**95**	1.21	65	1.54	**95**	**0.89**
Bowl stacking	**100**	0.61	**100**	0.60	**100**	**0.57**
Triple bowl stacking	90	1.44	75	**1.28**	**100**	1.36
Cube stacking	5	1.65	5	0.86	**100**	**0.78**
Average	73	1.23	61	1.07	**99**	**0.90**

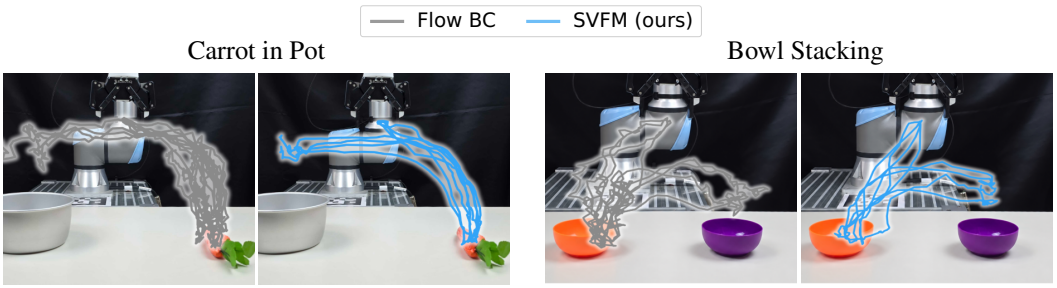


Figure 4: Successful end-effector trajectories on carrot in pot and bowl stacking, overlaid on a photograph of each scene for illustration. For each task, we show 4 successful trials per method from the same initial object placement. Figure 7 shows the trajectories in 3D.

Smoother and more efficient trajectories. Beyond task success, Figure 4 illustrates differences in motion quality among successful trials. Flow BC exhibits irregular, back-and-forth motion, whereas SVFM generally follows smoother, more consistent paths through grasping and release. The difference is visible even when both policies complete the task. Shorter paths, fewer gripper switches, and lower joint-velocity jitter across all trials (Table 17) suggest that critic-guided RL reduces unnecessary motion and promotes more efficient goal-reaching behavior. Together with cube stacking, on which Flow BC never succeeds, these observations indicate that offline RL changes the behavior of the flow policy it starts from rather than only its success rate; since each policy is trained once, we regard them as qualitative evidence (Appendix J.2).

5 Related Work

Offline RL with diffusion and flow policies. Offline RL must control extrapolation from limited data: behavior constraints and conservative value learning keep policy improvement near the data distribution (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2019; Kumar et al., 2020; Fujimoto & Gu, 2021), while IQL (Kostrikov et al., 2022) learns values without querying out-of-distribution actions. Diffusion policies model multimodal actions and can be optimized through the denoising process (Wang et al., 2023; Janner et al., 2022). Flow matching models actions through a learned velocity field (Lipman et al., 2023; Liu et al., 2023b), motivating flow-based offline RL (Park et al., 2025; Doo et al., 2026). One approach improves these policies through weighted behavior cloning (Peng et al., 2019; Nair et al., 2020; Zhang et al., 2025). Another selects from behavior-model samples or steers at test time (Chen et al., 2023; Hansen-Estruch et al., 2023; Nakamoto et al., 2024; Mao et al., 2024; Li et al., 2024; Ruan et al., 2026). Approximate action generation, one-step policy extraction, and score regularization avoid trajectory BPTT (Kang et al., 2023; Park et al., 2025; Chen et al., 2024). Other methods train generative policies with local supervision (Psenka et al., 2023; Fang et al., 2025; Alles et al., 2025). More broadly, entropy-regularized control views generative-model fine-tuning as a reward tilt of a reference path law (Uehara et al., 2024; Tang, 2024; Uehara et al., 2025); the Doob h-transform expresses its local guidance through a conditional soft value (Doob, 1957; Chen et al., 2021). Adjoint matching targets the KL-regularized optimal policy through SoC (Domingo-Enrich et al., 2025; Li & Levine, 2026). SVFM shares CEP's log-partition principle (Lu et al., 2023), but learns soft values from behavior continuations at a shared anchor and trains an actor without test-time guidance. Appendix L details these differences.

6 Conclusion

SVFM trains a multi-step actor with soft-value guidance from a fixed behavior model, avoiding trajectory BPTT while keeping the critic-query distribution for policy supervision independent of the current actor. It achieves the highest average score in both OGBench settings and supports offline-to-online learning and VLA fine-tuning for simulated and real-world manipulation.

Limitations. Performance depends on critic accuracy and behavior coverage. Finite-K soft-value estimation trades accuracy for computation and adds training cost (Appendix M.5); deployment uses only the trained actor. Real-world evaluation covers four tasks with 20 trials each, requiring broader validation across tasks, environments, and robot platforms.

AI USE STATEMENT

We used generative AI tools to assist with writing, grammar correction and language polishing, literature retrieval and discovery, refinement of research hypotheses, code implementation, and mathematical proof writing. We also used these tools to assist with figure and table preparation.

ETHICS STATEMENT

This work does not involve human-subject research or personal data. The simulated experiments use the publicly available OGBench, LIBERO, and LIBERO-Plus benchmarks, and our VLA experiments fine-tune publicly released pretrained models ($\pi_{0.5}$ and GR00T N1.7) in accordance with their licenses. Real-world demonstrations were collected by the authors through teleoperation, and the cameras recorded only the robot workspace, with no people in view. All real-robot experiments were conducted in a controlled laboratory setting with an emergency stop available. Because SVFM improves policies using a learned critic, its behavior can still be affected by critic errors, and policies trained with offline RL should be validated under supervision before deployment near people. We do not foresee applications of this work beyond general-purpose robot learning that raise specific ethical concerns.

REPRODUCIBILITY STATEMENT

We describe our method in Section 3 and provide implementation details, experimental configurations, hyperparameter selection procedures, and evaluation protocols in Appendices E, F, G, and K. We also provide proofs and assumptions for our theoretical results in the appendix, together with details of the real-world hardware, demonstration collection, and policy training in Appendix J.1. The hyperparameters and settings of all experiments, including the OGBench, offline-to-online, LIBERO and LIBERO-Plus, and real-world experiments, are reported in the appendix. We plan to publicly release our code.

REFERENCES

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. (Cited on pages 3 and 22.)

Marvin Alles, Nutan Chen, Patrick van der Smagt, and Botond Cseke. FlowQ: Energy-guided flow policies for offline reinforcement learning. *arXiv preprint arXiv:2505.14139*, 2025. (Cited on pages 1, 2, 4, and 9.)

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. (Cited on pages 6, 8, and 35.)

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025. URL https://proceedings.mlr.press/v305/black25a.html. (Cited on pages 6, 8, and 45.)

Jongseong Chae, Jongeui Park, Yongjae Shin, Gyeongmin Kim, Seungyul Han, and Youngchul Sung. Flow actor-critic for offline reinforcement learning. In *ICLR*, 2026. (Cited on pages 26, 39, and 40.)

Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, et al. GR-3 technical report. *arXiv preprint arXiv:2507.15493*, 2025. (Cited on page 1.)

Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *ICLR*, 2023. URL https://arxiv.org/abs/2209.14548. (Cited on page 9.)

Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. In *ICLR*, 2024. URL https://arxiv.org/abs/2310.07297. (Cited on page 9.)

Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Stochastic control liaisons: Richard Sinkhorn meets Gaspard Monge on a Schrödinger bridge. *SIAM Review*, 63(2):249–313, 2021. (Cited on pages 2, 3, 4, 9, and 18.)

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023. (Cited on page 1.)

Paolo Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied Mathematics and Optimization*, 23:313–329, 1991. (Cited on page 1.)

Sanjit Dandapanthula and Nicholas M. Boffi. Are we really tilting? the mechanics of reward guidance in flow and diffusion models. *arXiv preprint arXiv:2606.02884*, 2026. URL https://arxiv.org/abs/2606.02884. (Cited on page 18.)

Z. Ding and C. Jin. Consistency models as a rich and efficient policy class for reinforcement learning. In *ICLR*, 2024. (Cited on page 25.)

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *ICLR*, 2025. (Cited on pages 3, 9, 18, 22, and 23.)

Perry Dong, Qiyang Li, Dorsa Sadigh, and Chelsea Finn. Expo: Stable reinforcement learning with expressive policies. In *International Conference on Learning Representations*, volume 2026, pp. 131379–131394, 2026. (Cited on page 26.)

JaeHyeok Doo, Byeongguk Jeon, Seonghyeon Ye, Kimin Lee, and Minjoon Seo. Q-Flow: Stable and expressive reinforcement learning with flow-based policy. In *ICML*, 2026. (Cited on pages 1, 2, 4, 7, 9, 25, 26, 27, 28, 29, 30, 32, 34, 37, and 43.)

Joseph L Doob. Conditional Brownian motion and the boundary limits of harmonic functions. *Bulletin de la Société Mathématique de France*, 85:431–458, 1957. (Cited on pages 2, 4, 9, and 18.)

Linjiajie Fang, Ruoxue Liu, Jing Zhang, Wenjia Wang, and Bingyi Jing. Diffusion actor-critic: Formulating constrained policy iteration as diffusion noise regression for offline reinforcement learning. In *ICLR*, 2025. (Cited on pages 1, 9, 25, and 26.)

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025. (Cited on pages 2, 6, and 45.)

Wendell H Fleming and Halil Mete Soner. *Controlled Markov processes and viscosity solutions*. Springer, 2006. (Cited on page 1.)

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning, 2020. URL https://arxiv.org/abs/2004.07219. (Cited on page 27.)

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *NeurIPS*, 2021. (Cited on pages 2, 5, and 9.)

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *ICML*, 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html. (Cited on pages 1, 2, and 9.)

Seyed Kamyar Seyed Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline RL through ensembles, and why their independence matters., 2022. URL https://openreview.net/forum?id=wQ7RCayXUSl. (Cited on page 26.)

Igor V. Girsanov. On transforming a certain class of stochastic processes by absolutely continuous substitution of measures. *Theory of Probability & Its Applications*, 5(3):285–301, 1960. doi: 10.1137/1105027. (Cited on page 3.)

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023. (Cited on pages 1, 9, and 25.)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. (Cited on page 8.)

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9. (Cited on page 35.)

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/janner22a.html. (Cited on page 9.)

J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30:175–193, 1906. doi: 10.1007/BF02418571. (Cited on pages 4 and 20.)

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. In *NeurIPS*, 2023. URL https://arxiv.org/abs/2305.20081. (Cited on page 9.)

Hilbert J Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11011, 2005. doi: 10.1088/1742-5468/2005/11/P11011. (Cited on page 1.)

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. URL https://arxiv.org/abs/1412.6980. (Cited on page 26.)

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *ICLR*, 2022. (Cited on pages 9, 25, 34, 35, and 46.)

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *NeurIPS*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c2073ffa77b5357a498057413bb09d3a-Abstract.html. (Cited on pages 1, 2, and 9.)

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html. (Cited on pages 2 and 9.)

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. URL https://arxiv.org/abs/2005.01643. (Cited on pages 1 and 2.)

Qiyang Li and Sergey Levine. Q-learning with adjoint matching. In *ICLR*, 2026. (Cited on pages 1, 2, 3, 6, 9, 25, 26, 27, 28, 31, 32, 43, and 44.)

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=XUks1Y96NR. (Cited on page 26.)

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024. (Cited on pages 1, 4, and 9.)

Jia Lin Hau, Marek Petrik, and Mohammad Ghavamzadeh. Entropic risk optimization in discounted MDPs. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 47–76, 2023. URL https://proceedings.mlr.press/v206/lin-hau23a.html. (Cited on page 19.)

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. (Cited on pages 3, 9, and 21.)

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023a. (Cited on pages 2, 6, and 45.)

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023b. (Cited on pages 3 and 9.)

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *ICML*, 2023. (Cited on pages 1, 9, 26, 27, 38, 43, and 44.)

Liyuan Mao, Haoran Xu, Xianyuan Zhan, Weinan Zhang, and Amy Zhang. Diffusion-DICE: In-sample diffusion guidance for offline reinforcement learning. In *NeurIPS*, 2024. URL https://arxiv.org/abs/2407.20109. (Cited on page 9.)

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020. (Cited on pages 1, 2, and 9.)

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. *arXiv preprint arXiv:2410.13816*, 2024. (Cited on page 9.)

Gary W. Oehlert. A note on the delta method. *The American Statistician*, 46(1):27–29, 1992. doi: 10.1080/00031305.1992.10475842. (Cited on page 19.)

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. *arXiv preprint arXiv:2410.20092*, 2024. (Cited on pages 2 and 6.)

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. *arXiv preprint arXiv:2502.02538*, 2025. (Cited on pages 1, 2, 6, 9, 25, 26, 27, 29, 30, 32, 43, and 44.)

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. (Cited on pages 1, 2, and 9.)

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *AAAI*, 2010. (Cited on page 1.)

Physical Intelligence. openpi: Open-source models and code for the π_0 and $\pi_{0.5}$ vision-language-action models. https://github.com/Physical-Intelligence/openpi, 2025. (Cited on page 45.)

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via Q-score matching. *arXiv preprint arXiv:2312.11752*, 2023. (Cited on pages 1, 2, 4, 9, and 25.)

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. FLOWER: Democratizing generalist robot policies with efficient vision-language-flow models. In *CoRL*, 2025. (Cited on page 1.)

Yifan Ruan, Chenyang Cao, Andreas Burger, Ali Pesaranghader, Kaveh Kamali, Jaehong Kim, Nandita Vijaykumar, Alán Aspuru-Guzik, Igor Gilitschenski, and Nicholas Rhinehart. QPILOTS: Efficient test-time Q-steering for flow policies. *arXiv preprint arXiv:2606.14801*, 2026. (Cited on pages 1 and 9.)

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, et al. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. (Cited on page 1.)

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. (Cited on pages 3, 4, 18, and 22.)

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT Press, 2 edition, 2018. URL https://mitpress.mit.edu/9780262039246/reinforcement-learning/. (Cited on page 2.)

Wenpin Tang. Fine-tuning of diffusion models via stochastic control: Entropy regularization and beyond. *arXiv preprint arXiv:2403.06279v1*, 2024. URL https://arxiv.org/abs/2403.06279v1. (Cited on pages 3 and 9.)

D. Tarasov, V. Kurenkov, A. Nikulin, and S. Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *NeurIPS*, 2023. (Cited on page 25.)

Jan Tauberschmidt, Sophie Fellenz, Sebastian J. Vollmer, and Andrew B. Duncan. Physics-constrained fine-tuning of flow-matching models for generation and inverse problems. In *ICLR*, 2026. URL https://arxiv.org/abs/2508.09156. (Cited on page 18.)

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024. (Cited on pages 2, 3, 4, 9, 18, and 23.)

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025. (Cited on pages 2, 4, and 9.)

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. doi: 10.1561/2200000001. URL https://doi.org/10.1561/2200000001. (Cited on page 24.)

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *ICLR*, 2023. (Cited on pages 1, 9, and 25.)

Ziqian Wang, Rui Zhang, Yitian Liu, Xingjian Mao, Minqian Wang, and Yao Mu. Q-VGM: Q-guided value-gradient matching for offline-to-online RL of flow-matching VLA policies. *arXiv preprint arXiv:2606.08015*, 2026. URL https://arxiv.org/abs/2606.08015. (Cited on pages 8, 45, and 46.)

Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024. (Cited on page 34.)

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019. URL https://arxiv.org/abs/1911.11361. (Cited on pages 2 and 9.)

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *ICLR*, 2025. (Cited on page 9.)

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, et al. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025. (Cited on page 1.)

Zhiyuan Zhou, Andy Peng, Charles Xu, Qiyang Li, Tobias Springenberg, Kevin Frans, and Sergey Levine. Test-time gradient guidance of flow policies in reinforcement learning. *arXiv preprint arXiv:2606.11087*, 2026. (Cited on pages 8 and 46.)

Appendix Overview

The appendices are grouped by topic: proofs and theoretical analysis of guidance and adaptive temperature (Appendices A–D); implementation, experimental settings, and hyperparameter selection (Appendices E–G); per-task results and critic-query diagnostics (Appendices H–I); real-world and offline-to-online experiments and the CEP/QGPO comparison (Appendices J–L); and the FAC-protocol comparison, continuation-source analysis, wall-clock timing, and LIBERO/LIBERO-Plus details (Appendices M–N).

Appendix Contents

A Proofs and Finite-Sample Bias Analysis **18**

A.1 Proof of Proposition 1 . . . 18

A.2 Proof of Proposition 2 . . . 19

A.3 Bias of the Finite-Sample Soft-Value Target . . . 19

B Regularity Conditions and Actor Regression **21**

B.1 Pointwise minimizer of the actor objective . . . 21

C Early-Time Behavior of the Soft-Value Guidance **22**

C.1 Conditional endpoint law of the memoryless process . . . 22

C.2 Soft-value gradient and the early-time limit . . . 22

D Details of Adaptive Temperature **23**

D.1 Calibrating the control cost . . . 23

D.2 A KL bound for the target policy . . . 23

D.3 Trust-region interpretation and scope . . . 24

E Implementation Details **24**

F Experimental Settings **25**

G Hyperparameter Selection **27**

G.1 Standard Setting . . . 27

G.2 Advanced Setting . . . 28

G.3 Offline-to-Online Learning . . . 29

H Per-task Results **30**

I Critic Queries for Policy Supervision **32**

I.1 Toy Experiment . . . 32

I.2 OGBench . . . 33

J Real-World Experiment Details **33**

J.1 Experimental Setup . . . 33

J.2 Beyond Success Rate 35

K Details of the Offline-to-Online Experiment **37**

L Relation to CEP and QGPO **38**

M Additional Analysis **39**

M.1 Complementarity with Advanced Critic Learning 39

M.2 Behavior Continuations versus Actor Continuations 41

M.3 Sensitivity to the Guidance Cap and the Minimum Flow Time 41

M.4 Number of Continuations 41

M.5 Wall-Clock Training Time 42

N LIBERO and LIBERO-Plus Experimental Details **45**

A Proofs and Finite-Sample Bias Analysis

A.1 Proof of Proposition 1

We specialize the standard terminal-tilting and h-transform construction (Doob, 1957; Chen et al., 2021; Uehara et al., 2024) to the scaled memoryless reference process. Fix s, a bounded Q, $\lambda > 0$, and $\kappa \in (0, 1]$, keeping λ and κ constant over flow time. Assume the regularity and endpoint-limit conditions of Appendix B for this process; in particular, endpoint identities below are understood as boundary limits.

Scaled reference process. Let p_t be the behavior CFM marginal density and let u be its exact velocity. Set

$$\sigma_{\kappa,t}^2 = 2\kappa^2 \frac{1-t}{t}, \qquad b_\kappa(x,t) = u(x,t) + \frac{\sigma_{\kappa,t}^2}{2} \nabla_x \log p_t(x) = u(x,t) - \kappa^2 \frac{x - t\,u(x,t)}{t}.$$

The last equality uses the linear-interpolation score identity derived in Appendix B. The SDE with drift b_κ and diffusion $\sigma_{\kappa,t} I$ is equation 10 with $u_\phi = u$; denote its path law by P_κ. Its Fokker–Planck equation reduces to the CFM continuity equation, so it has the same marginals p_t, with $p_0 = \mathcal{N}(0, I)$ and $p_1 = \pi_\beta(\cdot \mid s)$. Moreover, X_0 and X_1 remain independent for every fixed $\kappa \in (0, 1]$. This is the scaled-memoryless result of Tauberschmidt et al. (2026, Appendix D.4, Lemma 1), with their scaling parameter set to $1 - \kappa^2 \in [0, 1)$; $\kappa = 1$ recovers the original memoryless process (Domingo-Enrich et al., 2025).

Terminal tilt and initial law. Define the soft value using this same reference law:

$$h_\kappa(x,t) := \mathbb{E}_{P_\kappa}\left[e^{Q(s,X_1)/\lambda} \mid X_t = x \right], \qquad V_{\lambda,\kappa}(s,x,t) := \lambda \log h_\kappa(x,t),$$
$$Z := \mathbb{E}_{a \sim \pi_\beta(\cdot|s)}\left[e^{Q(s,a)/\lambda} \right].$$

Boundedness of Q ensures $0 < Z < \infty$. Endpoint independence gives

$$h_\kappa(x,0) = \mathbb{E}_{P_\kappa}\left[e^{Q(s,X_1)/\lambda} \mid X_0 = x \right] = Z,$$

independently of x. The terminally tilted path law $dP_\kappa^*/dP_\kappa = e^{Q(s,X_1)/\lambda}/Z$ therefore has marginals

$$p_{\kappa,t}^*(x) = \frac{p_t(x) h_\kappa(x,t)}{Z}. \tag{12}$$

It starts from the same $\mathcal{N}(0, I)$ law, and ends at $p_{\kappa,1}^*(a) = \pi_\beta(a \mid s) e^{Q(s,a)/\lambda}/Z = \pi^\star(a \mid s)$. For this terminal-tilting identity and its proof, see Dandapanthula & Boffi (2026, Theorem 12); their exponent λr corresponds to our Q/λ. Because the tilt depends only on X_1, it leaves the conditional path law given X_1 unchanged. The chain rule for relative entropy then reduces the path-space KL to the terminal KL (Chen et al., 2021; Domingo-Enrich et al., 2025), so this feasible path law attains the action-space optimum equation 1.

Probability-flow velocity. The h-transform adds the drift $c_\kappa^* = \sigma_{\kappa,t}^2 \nabla_x \log h_\kappa$. Applying the standard probability-flow conversion (Song et al., 2021, Eq. (13), Appendix D.1) gives

$$\begin{aligned} v_\kappa^*(x,t) &= b_\kappa(x,t) + c_\kappa^*(x,t) - \frac{\sigma_{\kappa,t}^2}{2} \nabla_x \log p_{\kappa,t}^*(x) \\ &= u(x,t) + \frac{\sigma_{\kappa,t}^2}{2} \nabla_x \log h_\kappa(x,t) \\ &= u(x,t) + \kappa^2 \frac{1-t}{t\lambda} \nabla_x V_{\lambda,\kappa}(s,x,t). \end{aligned} \tag{13}$$

The factor $1/2$ comes from the probability-flow score correction. The ODE shares the controlled SDE's time marginals, rather than its path law. Under the assumed endpoint limits, this ODE transports $p_0 = \mathcal{N}(0, I)$ to $p_{\kappa,1}^* = \pi^\star$, proving Proposition 1. Setting $\kappa = 1$ recovers equation 8, equivalently the guided probability-flow formula of Dandapanthula & Boffi (2026, Eq. (4)).

Scope of the result. The terminal optimum is independent of $\kappa \in (0, 1]$ for fixed Q, λ, using the same κ in the soft value and ODE coefficient. The endpoint-independence argument does not cover $\kappa = 0$. This fixed-objective result does not guarantee convergence of adaptive training with changing Q and update-wise λ_n as described in Appendix E.

A.2 Proof of Proposition 2

The proof applies the translation and joint-rescaling properties of the entropic log-exponential functional (Lin Hau et al., 2023, Theorem 3.1 and Definition A.1) to our conditional soft value equation 7.[1] We specialize these properties to the adaptive-temperature rule equation 11 and the guidance coefficient below.

We keep the continuation law P_κ, the anchor distribution, and (κ, g) fixed, and let $Q' = cQ + b(s)$ with $c > 0$. At each anchor, $b(s)$ is constant across continuation endpoints, while multiplying Q by $c > 0$ multiplies its standard deviation by c. Averaging over the same anchors therefore gives

$$s_{Q'} = c\, s_Q, \qquad \lambda' = \frac{\kappa^2 s_{Q'}}{g} = c\lambda.$$

The same identities hold for empirical estimates computed on the same anchors and continuation endpoints. The assumption $0 < s_Q < \infty$ ensures that both temperatures are finite and strictly positive.

Let V and V' be the corresponding finite exact soft values, differentiable in x. Since $b(s)$ is constant under the conditional expectation, their definitions directly give

$$\begin{aligned} V'(s, x, t) &= c\lambda \log \mathbb{E}_{P_\kappa}\left[e^{(cQ(s,X_1)+b(s))/(c\lambda)} \mid X_t = x\right] \\ &= cV(s, x, t) + b(s). \end{aligned}$$

Differentiating with s, λ, and λ' fixed gives $\nabla_x V' = c\nabla_x V$. Since $\beta_t^{(\kappa)'} = \beta_t^{(\kappa)}/c$, the factors cancel:

$$\beta_t^{(\kappa)'} \nabla_x V' = \frac{\beta_t^{(\kappa)}}{c}\, c\nabla_x V = \beta_t^{(\kappa)} \nabla_x V.$$

Since the CFM target $a - x_0$ is unchanged, the complete sample-level actor target $(a - x_0) + \beta_t^{(\kappa)} \nabla_x V$ is unchanged as well.

The same algebra applied to the finite-sample log-mean-exp target gives $\widehat{V}'_K = c\widehat{V}_K + b(s)$ for the same continuation samples. The proposition concerns the exact guidance under a fixed continuation law and the unmodified adaptive-temperature rule, without numerical floors. It does not assert that finite-capacity networks, numerical optimization, or the complete TD learning dynamics are invariant to reward transformations.

A.3 Bias of the Finite-Sample Soft-Value Target

We apply a standard Taylor/delta-method argument for transformed sample means (Oehlert, 1992) to the log-mean-exp target used by SVFM. We give the boundedness assumptions and the finite-K remainder explicitly.

Assumptions and well-definedness. Fix an anchor (s, x_t, t), a critic Q, a temperature $\lambda > 0$, and the continuation law. All expectations and moments in this section are conditional on these fixed quantities. Assume the continuation endpoints $a^{(1)}, \ldots, a^{(K)}$ are conditionally independent and identically distributed, and that

$$|Q(s, a)| \le M_Q < \infty$$

on the support of the relevant endpoint laws. This is an explicit assumption for the analysis, not a boundedness guarantee for an unconstrained neural critic. Set

$$Z_j := e^{Q(s,a^{(j)})/\lambda}, \qquad \bar{Z}_K := \frac{1}{K}\sum_{j=1}^{K} Z_j, \qquad \mu := \mathbb{E}[Z_1], \qquad \nu_r := \mathbb{E}[(Z_1 - \mu)^r] \quad (r = 2, 3, 4).$$

[1] In their sign convention, our soft value is $V_\lambda(Q) = -\,\mathrm{ERM}_{1/\lambda}[-Q]$, where ERM denotes their entropic risk measure, here evaluated under the fixed conditional continuation law.

In particular, $\nu_2 = \mathrm{Var}(Z_1)$. With $m := e^{-M_Q/\lambda} > 0$ and $M := e^{M_Q/\lambda} < \infty$, we have $m \le Z_j, \bar{Z}_K, \mu \le M$. Thus the exponential expectation is finite and strictly positive, the logarithms and all moments used below exist, and the soft value $V := \lambda \log \mu$ is finite. For the reference law P_β, $V = V_\lambda(s, x_t, t)$; the same calculation applies to $V_{\lambda,\kappa}$ when the endpoints follow P_κ.

The regression target is

$$\widehat{V}_K := \lambda \log \bar{Z}_K = \lambda \log \left(\frac{1}{K} \sum_{j=1}^{K} e^{Q(s,a^{(j)})/\lambda} \right).$$

Both $\widehat{V}_K$ and V lie in $[-M_Q, M_Q]$. Before applying the logarithm, the sample mean is unbiased:

$$\mathbb{E}[\bar{Z}_K] = \mu.$$

Concavity of the logarithm then gives

$$\mathbb{E}[\widehat{V}_K] = \lambda \mathbb{E}[\log \bar{Z}_K] \le \lambda \log \mu = V.$$

Moreover, the strong law gives $\bar{Z}_K \to \mu$ almost surely, and bounded convergence yields $\mathbb{E}[\widehat{V}_K] \to V$.

Second-order expansion and remainder. Write $\Delta_K := \bar{Z}_K - \mu$. Conditional independence implies

$$\mathbb{E}[\Delta_K] = 0, \qquad \mathbb{E}[\Delta_K^2] = \frac{\nu_2}{K}, \qquad \mathbb{E}[\Delta_K^4] = \frac{\nu_4 + 3(K-1)\nu_2^2}{K^3}.$$

Taylor's theorem with the Lagrange remainder gives

$$\log(\mu + \Delta_K) = \log \mu + \frac{\Delta_K}{\mu} - \frac{\Delta_K^2}{2\mu^2} + R_{K,3}.$$

For each realization, there is a ξ_K between μ and $\bar{Z}_K$. Since $(\log z)''' = 2/z^3$ and both endpoints lie in $[m, M]$,

$$R_{K,3} = \frac{(\log z)'''|_{z=\xi_K}}{3!} \Delta_K^3 = \frac{\Delta_K^3}{3\xi_K^3}, \qquad \xi_K \ge m, \qquad |R_{K,3}| \le \frac{|\Delta_K|^3}{3m^3}.$$

Because $\mathbb{E}[\Delta_K] = 0$, the linear term vanishes after taking the conditional expectation, not sample by sample. To bound the expected remainder, the fourth-moment identity above gives, for $K \ge 1$,

$$\mathbb{E}[\Delta_K^4] = \frac{\nu_4}{K^3} + \frac{3(K-1)\nu_2^2}{K^3} \le \frac{C_4}{K^2}, \qquad C_4 := \nu_4 + 3\nu_2^2,$$

where C_4 is independent of K. Jensen's inequality (Jensen, 1906) applied to the concave function $y \mapsto y^{3/4}$ on $[0, \infty)$ then yields

$$\mathbb{E}|\Delta_K|^3 = \mathbb{E}\big[(\Delta_K^4)^{3/4}\big] \le \big(\mathbb{E}[\Delta_K^4]\big)^{3/4} \le \left(\frac{C_4}{K^2}\right)^{3/4} = C_4^{3/4} K^{-3/2}.$$

Thus $\mathbb{E}|\Delta_K|^3 = O(K^{-3/2})$. With $r_K := \lambda \mathbb{E}[R_{K,3}]$, the bias of the full soft-value target satisfies

$$\mathbb{E}[\widehat{V}_K] - V = -\frac{\lambda \nu_2}{2K\mu^2} + r_K,$$
$$|r_K| \le \frac{\lambda}{3m^3} \mathbb{E}|\Delta_K|^3 = O(K^{-3/2}).$$

Thus the leading bias is of order $1/K$ when $\nu_2 > 0$. If $\nu_2 = 0$, Z_1 is constant almost surely and the bias is exactly zero.

Sharper expected-bias bound. Boundedness also permits one more Taylor term:

$$\log(\mu + \Delta_K) = \log \mu + \frac{\Delta_K}{\mu} - \frac{\Delta_K^2}{2\mu^2} + \frac{\Delta_K^3}{3\mu^3} + R_{K,4}, \qquad |R_{K,4}| \le \frac{\Delta_K^4}{4m^4}.$$

Using $\mathbb{E}[\Delta_K^3] = \nu_3/K^2$ and the fourth-moment identity above gives

$$\begin{aligned}\mathbb{E}[\widehat{V}_K] - V &= -\frac{\lambda\nu_2}{2K\mu^2} + \frac{\lambda\nu_3}{3K^2\mu^3} + \lambda\mathbb{E}[R_{K,4}] \\ &= -\frac{\lambda\nu_2}{2K\mu^2} + O(K^{-2}),\end{aligned}$$

because

$$|\lambda\mathbb{E}[R_{K,4}]| \le \frac{\lambda}{4m^4}\frac{\nu_4 + 3(K-1)\nu_2^2}{K^3} = O(K^{-2}).$$

The $O(K^{-2})$ statement concerns the expected bias; it is not a samplewise remainder bound for the second-order expansion.

Scope under adaptive temperature. All rates are for fixed Q, $\lambda > 0$, and continuation law as $K \to \infty$. Their constants depend on M_Q and λ and need not remain bounded as $\lambda \downarrow 0$. Adaptive temperature controls the critic scale used in the target, but does not by itself guarantee bounded Q or finite exponential moments. These results do not establish convergence of training with a changing critic or temperature.

B Regularity Conditions and Actor Regression

We state the regularity and endpoint-limit assumptions used in Appendix A.1 and connect its probability-flow velocity to the actor objective. The controlled marginals and velocity are given by equation 12 and equation 13. These assumptions are imposed separately for each fixed $\kappa \in (0, 1]$. For the assumptions below, write $h = h_\kappa$ and $\sigma_t = \sigma_{\kappa,t}$.

We fix s, Q, and $\lambda > 0$ and assume the bounded-critic condition of Appendix A.3. This ensures $0 < \mathbb{E}_{P_\kappa}[\exp(Q(s, X_1)/\lambda)] < \infty$ and finite positive conditional h-values. Separately, we work under sufficient smoothness, positivity of the time-marginal densities, and integrability for the conditional expectations and probability-flow construction. The diffusion is scalar and state-independent, $\sigma_t I$. We assume sufficient boundary decay and well-posed, nonexplosive dynamics so that the Fokker–Planck and continuity equations determine their marginals uniquely. The SDE dynamics are understood on the open interval $0 < t < 1$. All endpoint identities below denote boundary limits. The probability-flow construction is applied on compact subintervals $[\varepsilon, 1-\varepsilon]$; we assume these flows admit the stated endpoint limits as $\varepsilon \downarrow 0$. Here p_t denotes the P_κ-marginal density, with $p_0 = \mathcal{N}(0, I)$ and $p_1 = \pi_\beta(\cdot \mid s)$, and h is finite, positive, and sufficiently differentiable on the open interval.

B.1 Pointwise minimizer of the actor objective

For the actor-regression calculation below, we specialize to $\kappa = 1$, writing $h = h_1$ and $V_\lambda = V_{\lambda,1}$. Let $x_t = (1-t)x_0 + ta$, where $x_0 \sim \mathcal{N}(0, I)$ and $a \sim \pi_\beta(\cdot \mid s)$ are independent conditional on s. Under this anchor law, the marginal-field identity and regression equivalence of conditional flow matching (Lipman et al., 2023, Eq. (8), Theorem 2) give $\mathbb{E}[a - x_0 \mid s, x_t = x, t] = u(x, t)$ for $0 < t < 1$. Differentiating the Gaussian conditional density $p_t(x \mid a) = \mathcal{N}(x; ta, (1-t)^2 I)$ also gives

$$\nabla_x \log p_t(x) = -\frac{\mathbb{E}[x_0 \mid s, x_t = x, t]}{1-t} = -\frac{x - t\,u(x,t)}{1-t},$$

the score identity used in Appendix A.1. For a fixed differentiable V and coefficient β_t, define $Y = (a - x_0) + \beta_t \nabla_x V(s, x_t, t)$. Assuming a finite conditional second moment, squared-error regression over unrestricted measurable velocity fields is minimized by the conditional mean, almost everywhere under the anchor law. Since the guidance term is fixed given (s, x_t, t),

$$\begin{aligned}v^\star_{\mathrm{reg}}(x, t) &= \mathbb{E}[Y \mid s, x_t = x, t] \\ &= u(x, t) + \beta_t \nabla_x V(s, x, t).\end{aligned}$$

Thus $V = V_{\lambda,1} = V_\lambda$ and $\beta_t = (1-t)/(t\lambda)$ give $v^\star_{\mathrm{reg}} = v^\star_1$ from equation 13 in Appendix A.1, recovering equation 8 and connecting the actor loss equation 9 to the probability-flow velocity. The stop-gradient sg keeps Y fixed during actor differentiation and does not change this population minimizer. Learned V_ψ, finitely many continuations, finite actor capacity, and incomplete optimization introduce approximation error, so this identity alone does not establish exactness of the finite implementation.

C Early-Time Behavior of the Soft-Value Guidance

We analyze the memoryless reference process equation 4 (Domingo-Enrich et al., 2025) at $\kappa = 1$, with fixed s, Q, and $\lambda > 0$ and the exact behavior field u. We assume the regularity conditions for SDE time reversal, $\mathbb{E}_{a\sim\pi_\beta}\|a\| < \infty$, and $\mathbb{E}_{a\sim\pi_\beta}[(1+\|a\|)e^{Q(s,a)/\lambda}] < \infty$. These moment conditions ensure a finite positive tilting normalizer and finite first moments under both π_β and $\pi^\star$. All limits below are for fixed finite x. The SDE is defined on $0 < t < 1$, with endpoint laws understood as boundary limits, without asserting a pathwise-continuous extension at $t = 0$. We use density notation; the tilting identities also apply to a general behavior measure.

C.1 Conditional endpoint law of the memoryless process

Introduce the reverse clock $\tau = 1 - t$ and $Y_\tau = X_{1-\tau}$. Using the score identity equation 3 (Albergo et al., 2023), the standard time-reversal formula (Song et al., 2021) applied to equation 4 gives

$$dY_\tau = -\frac{Y_\tau}{1-\tau}\,d\tau + \sqrt{\frac{2\tau}{1-\tau}}\,dB_\tau, \qquad Y_0 = a,$$

where B_τ is Brownian motion in the reversed clock. Conditional on $Y_0 = a$, this linear SDE with deterministic coefficients is Gaussian. Solving its mean and covariance equations gives $(1-\tau)a$ and $\tau^2 I$, respectively, hence $X_t \mid X_1 = a \sim \mathcal{N}(ta, (1-t)^2 I)$. Bayes' rule then gives the conditional endpoint law ρ_t and its Q-tilted counterpart:

$$\begin{aligned} L_t(x,a) &:= \exp\left(-\frac{\|x-ta\|^2}{2(1-t)^2}\right), \\ \rho_t(a \mid x) &\propto \pi_\beta(a \mid s) L_t(x,a), \\ \rho^\star_t(a \mid x) &\propto \rho_t(a \mid x) e^{Q(s,a)/\lambda}. \end{aligned}$$

C.2 Soft-value gradient and the early-time limit

The h-function is the ratio of the normalizing integrals of $\pi_\beta(a \mid s)L_t(x,a)e^{Q(s,a)/\lambda}$ and $\pi_\beta(a \mid s)L_t(x,a)$. Since $\nabla_x \log L_t(x,a) = -(x-ta)/(1-t)^2$, differentiating $\log h$ subtracts the corresponding conditional expectations of this score, cancelling the common $-x/(1-t)^2$ term. The moment assumptions justify differentiation under the integrals. Using $V_\lambda = \lambda \log h$ yields the guidance directly:

$$\frac{1-t}{t\lambda}\nabla_x V_\lambda(s,x,t) = \frac{\mathbb{E}_{\rho^\star_t(\cdot|x)}[a] - \mathbb{E}_{\rho_t(\cdot|x)}[a]}{1-t}.$$

For fixed finite x, $0 < L_t(x,a) \le 1$ and $L_t(x,a) \to e^{-\|x\|^2/2} > 0$ as $t \downarrow 0$. Dominated convergence, using the first-moment conditions above, gives $\rho_t(\cdot \mid x) \to \pi_\beta(\cdot \mid s)$ and $\rho^\star_t(\cdot \mid x) \to \pi^\star(\cdot \mid s)$, including convergence of their conditional means. Therefore,

$$\lim_{t\downarrow 0} \frac{1-t}{t\lambda}\nabla_x V_\lambda(s,x,t) = \mathbb{E}_{a\sim\pi^\star(\cdot|s)}[a] - \mathbb{E}_{a\sim\pi_\beta(\cdot|s)}[a].$$

Also, $h(x,t) \to \mathbb{E}_{a\sim\pi_\beta}[e^{Q(s,a)/\lambda}]$ and $\nabla_x V_\lambda(s,x,t) = O(t)$. The $O(1/t)$ guidance coefficient leaves the finite, x-independent mean shift above, which can be nonzero. At finite t, the conditional laws and guidance can still depend on the anchor.

Scope. These conditional-law and guidance identities are established for $\kappa = 1$; applying them to another κ requires a separate derivation. Taking $\kappa \downarrow 0$ at fixed $t > 0$ is a different limit from the $t \downarrow 0$ analysis here.

D Details of Adaptive Temperature

We relate the temperature rule equation 11 to the control cost in equation 5 and derive a statewise KL bound for the Gibbs target equation 1. Throughout, we fix the critic Q, the behavior continuation process, $\kappa \in (0, 1]$, and $g > 0$. The scalar $0 < s_Q < \infty$ is held fixed over flow time and with respect to the control being optimized. This is possible because its continuations come from the behavior process; its detached implementation is described in Appendix E.

D.1 Calibrating the control cost

Let b_κ denote the reference drift in equation 10 and let $\sigma^2_{\kappa,t} = 2\kappa^2(1-t)/t$. Write $P_{\kappa,c}$ for the path law obtained by adding a drift control c:

$$dX_t = [b_\kappa(X_t, t) + c(X_t, t)]dt + \sigma_{\kappa,t}\, dW_t.$$

The reference law P_κ has $c = 0$, the same initial law, and the same diffusion coefficient. Under the Girsanov conditions of equation 6, with endpoint limits understood as in Appendix B,

$$\begin{aligned} D_{\mathrm{KL}}(P_{\kappa,c} \,\|\, P_\kappa) &= \mathbb{E}_{\kappa,c}\left[\int_0^1 \frac{\|c(X_t,t)\|^2}{2\sigma^2_{\kappa,t}}\, dt\right] \\ &= \frac{1}{4\kappa^2}\mathbb{E}_{\kappa,c}\left[\int_0^1 \frac{t}{1-t}\|c(X_t,t)\|^2\, dt\right]. \end{aligned} \tag{14}$$

This compares controlled and reference processes at the same κ; it does not compare path laws with different diffusion coefficients. The identity is the standard drift-energy representation of path KL (Domingo-Enrich et al., 2025; Uehara et al., 2024).

Define $J_\kappa(c) = \mathbb{E}_{\kappa,c}[Q(s, X_1)] - \lambda D_{\mathrm{KL}}(P_{\kappa,c} \,\|\, P_\kappa)$. Dividing by the positive scalar s_Q preserves its maximizers and gives

$$\frac{J_\kappa(c)}{s_Q} = \mathbb{E}_{\kappa,c}\left[\frac{Q(s,X_1)}{s_Q} - \frac{\lambda}{4\kappa^2 s_Q}\int_0^1 \frac{t}{1-t}\|c(X_t,t)\|^2\, dt\right].$$

We parameterize the coefficient of this weighted drift effort as $1/(4g)$. This calibration yields

$$\frac{\lambda}{4\kappa^2 s_Q} = \frac{1}{4g} \quad\Longleftrightarrow\quad \lambda = \frac{\kappa^2 s_Q}{g},$$

recovering equation 11. Thus, s_Q normalizes the critic scale and κ^2 compensates for the explicit noise dependence of the drift-energy coefficient in equation 14. The dynamics, the expectation under $P_{\kappa,c}$, and s_Q still depend on κ; the calibration does not make the entire objective or its optimizer independent of κ.

We use the average within-anchor standard deviation of terminal critic values as the scale s_Q. It measures the value differences among the actions compared in soft-value estimation and is unaffected by an additive state-dependent baseline. Its population counterpart is the mean of the conditional standard deviations under the behavior continuations, rather than the standard deviation of all terminal values pooled across anchors. Changing κ changes these conditional distributions, so this dependence remains implicit in s_Q. The choice of this scale is a design choice; the algebra above holds for any fixed positive scalar scale.

D.2 A KL bound for the target policy

Fix an environment state s and let m_s and M_s be the essential infimum and supremum of $Q(s, \cdot)$ under $\pi_\beta(\cdot \mid s)$. Assume both are finite and define $R_s = M_s - m_s$. Here we impose the bounded-critic condition of Appendix A.3 on the support of $\pi_\beta(\cdot \mid s)$. Consider the ideal Gibbs target

$$\pi^\star_\lambda(a \mid s) = \frac{\pi_\beta(a \mid s)e^{Q(s,a)/\lambda}}{\mathbb{E}_{A\sim\pi_\beta(\cdot|s)}e^{Q(s,A)/\lambda}}, \qquad \lambda = \frac{\kappa^2 s_Q}{g}.$$

Proposition 3 (Statewise KL bound)**.** *Under the assumptions above, the target policy satisfies*

$$D_{\mathrm{KL}}(\pi^\star_\lambda(\cdot \mid s) \,\|\, \pi_\beta(\cdot \mid s)) \le \bar{\varepsilon}(s) := \frac{R_s^2}{8\lambda^2} = \frac{g^2}{8\kappa^4}\left(\frac{R_s}{s_Q}\right)^2. \tag{15}$$

Proof. For a scalar $b \geq 0$, define the log-partition function and tilted policy

$$A_s(b) = \log \mathbb{E}_{A \sim \pi_\beta(\cdot|s)} e^{bQ(s,A)}, \qquad \pi_b(a \mid s) = \pi_\beta(a \mid s) e^{bQ(s,a) - A_s(b)}.$$

For fixed s, Q, and π_β, these tilted policies form a one-parameter exponential family with natural parameter b, sufficient statistic $Q(s,\cdot)$, and base measure $\nu_s(da) = \pi_\beta(da \mid s)$. Their densities relative to ν_s are $e^{bQ(s,a)-A_s(b)}$; in particular, $\pi_0 = \pi_\beta$ and $\pi_{1/\lambda} = \pi^\star_\lambda$ belong to this same family. This construction does not require the behavior model itself to be parameterized as an exponential family. Boundedness of Q makes $A_s(b)$ finite for every $b \in \mathbb{R}$ and permits differentiation under the expectation. The standard cumulant identities therefore give $A'_s(b) = \mathbb{E}_{\pi_b}[Q]$ and $A''_s(b) = \mathrm{Var}_{\pi_b}(Q)$ (Wainwright & Jordan, 2008, Proposition 3.1). The exponential-family KL formula, applied to parameters b and 0, gives $D_{\mathrm{KL}}(\pi_b \,\|\, \pi_\beta) = bA'_s(b) - A_s(b)$ (Wainwright & Jordan, 2008, Eq. (5.10)). Since $A_s(0) = 0$, differentiating this KL expression and integrating from 0 to $1/\lambda$ yields

$$D_{\mathrm{KL}}(\pi^\star_\lambda \,\|\, \pi_\beta) = \int_0^{1/\lambda} b \, \mathrm{Var}_{\pi_b}(Q) \, db. \tag{16}$$

Every tilted policy has the same essential critic range. Consequently,

$$\mathrm{Var}_{\pi_b}(Q) \leq \mathbb{E}_{\pi_b}\left[\left(Q - \frac{m_s + M_s}{2}\right)^2\right] \leq \frac{R_s^2}{4}.$$

Substitution into equation 16 yields $D_{\mathrm{KL}}(\pi^\star_\lambda \,\|\, \pi_\beta) \leq (R_s^2/4) \int_0^{1/\lambda} b \, db = R_s^2/(8\lambda^2)$, proving the result. □

Under the common bound $|Q(s,a)| \leq M_Q$, one may use $R_s \leq 2M_Q$ to obtain $\bar{\varepsilon}(s) \leq g^2 M_Q^2/(2\kappa^4 s_Q^2)$. For a desired radius $\varepsilon_0(s) > 0$ and $R_s > 0$, the sufficient condition

$$g \leq \frac{\kappa^2 s_Q}{R_s} \sqrt{8\varepsilon_0(s)}$$

ensures that the ideal target lies within that KL radius. If $R_s = 0$, the target equals the behavior policy and its KL is zero.

D.3 Trust-region interpretation and scope

Proposition 3 bounds the target's KL from π_β by $\bar{\varepsilon}(s)$, which need not be attained. Standard-deviation normalization alone does not bound R_s/s_Q, so (κ, g) do not specify a universal radius. In particular, s_Q need not equal the standard deviation under π_β or any tilted policy in equation 16.

With the continuation law, anchor distribution, and (κ, g) fixed, Appendix A.2 gives $s_{Q'} = \alpha s_Q$ and $\lambda' = \alpha\lambda$ for $Q'(s,a) = \alpha Q(s,a) + d(s)$, $\alpha > 0$. Empirical comparisons use the same anchors and continuation endpoints. Since $R'_s = \alpha R_s$, Gibbs normalization and equation 15 give

$$\pi^{\star\prime}_{\lambda'} = \pi^\star_\lambda, \qquad \bar{\varepsilon}'(s) = \bar{\varepsilon}(s).$$

The bound applies conditionally to the ideal target for any positive detached estimate of s_Q; its radius need not remain fixed across updates. Appendix E discusses the scope under adaptive training. Exact affine invariance requires homogeneous scaling of s_Q, which an active numerical floor can break.

E Implementation Details

The batch sizes and fixed SVFM settings are listed in Table 7. For soft-value and temperature estimation, we draw $t \sim \mathcal{U}(t_{\min}, 1)$; the actor draws $t \sim \mathcal{U}(0,1)$, because at inference the actor is integrated from $t = 0$ and must therefore be trained there. As an implementation choice, soft-value guidance is disabled for $t < t_{\min}$.

Minimum flow time. We use $t_{\min} = 0.1$ as the minimum flow time for soft-value and temperature estimation, avoiding numerical continuations near the singular reference-process coefficients at $t = 0$. The actor's guidance coefficient also scales as $1/t$, which can amplify errors in the learned soft-value gradient at small t. This is a numerical safeguard: for $\kappa = 1$, the exact guidance has a finite early-time limit, as shown in Appendix C. Actor training still samples the full flow-time interval, with no soft-value guidance below $t_{\min}$.

Number of continuations. We fix $K = 8$ continuations per anchor to limit training computation while estimating each target from multiple samples. Increasing K increases continuation-generation and critic-evaluation work; the finite-K bias analysis in Appendix A.3 does not establish that $K = 8$ yields negligible error. This computation is confined to training and does not change actor inference. Neither K nor $t_{\min}$ is tuned per environment. Appendices M.3 and M.4 report the sensitivity to $t_{\min}$, the guidance cap and K.

Estimating s_Q. At each update, we integrate the scaled behavior process equation 10 from each anchor to $t = 1$ to obtain K terminal actions, evaluate the target critic on them, and take the standard deviation over the K actions per anchor. The batch mean of those standard deviations is s_Q, and $\lambda = \kappa^2 s_Q / g$ is used for both the soft-value target and the actor coefficient in that same update. Two guards: s_Q uses the positive floor in Table 7 so that a state where every continuation has the same value cannot send the coefficient to infinity, and the update-wise scalar λ_n carries a stop-gradient: target construction and the actor coefficient treat it as fixed rather than differentiating through the scale estimator.

Fixed-objective theory versus adaptive training. We use the same λ_n for all flow times involved in an update; it may change at the next optimization update. Holding it fixed along flow time is the condition used by the fixed-objective derivation, not a guarantee of convergence across updates. The implemented soft-value network is $V_\psi(s, x, t)$ and does not take λ as an input. As the critic and λ_n change, this network is trained to track a changing regression target, rather than representing an exact family V_λ indexed by λ. Its gradient at update n therefore need not equal the gradient of the exact soft value for the current Q and λ_n. Proposition 1 describes the ideal population construction for fixed inputs; it does not imply exact policy improvement or convergence of this adaptive, finite-capacity implementation.

F Experimental Settings

This appendix collects the fixed settings for the OGBench experiments. The standard and advanced protocols follow Park et al. (2025) and Li & Levine (2026), respectively, as in the two-regime comparison of Doo et al. (2026). Table 6 lists their configurations; scores are comparable within a regime, not directly across regimes. Table 7 gives the shared SVFM settings, while Appendix G describes hyperparameter selection.

Datasets contain 1M transitions by default, 2M for humanoidmaze, and 3M for cube-triple. In the advanced setting, puzzle-4x4 and cube-quadruple use 100M transitions. We evaluate every 100K steps with 50 episodes per task and report mean success rates over the 800K, 900K, and 1M evaluations, averaged over the 5 tasks of each domain.

Baselines. We compare the following three policy families.

Gaussian or deterministic policies. **IQL** (Kostrikov et al., 2022) learns values from dataset actions through expectile regression and extracts a Gaussian policy using advantage-weighted behavior cloning. **ReBRAC** (Tarasov et al., 2023) trains a deterministic actor by maximizing critic values while penalizing deviations from dataset actions.

Diffusion and consistency policies. **IDQL** (Hansen-Estruch et al., 2023) learns a diffusion behavior policy and extracts improved actions by selecting among its samples using an implicit Q-learning critic. **CAC** (Ding & Jin, 2024) combines consistency-based behavior cloning with Q maximization to train an expressive policy that generates actions in few sampling steps. **DAC** (Fang et al., 2025) formulates KL-constrained policy improvement as diffusion noise regression, combining behavior regularization with critic-gradient guidance. **QSM** (Psenka et al., 2023) trains a diffusion policy by matching its score to critic action gradients, with an additional behavior-cloning loss in the advanced-setting variant used by Li & Levine (2026).

Flow policies. **FBRAC** (Park et al., 2025) combines flow-matching with Q maximization through the full flow path, requiring trajectory BPTT (Wang et al., 2023). **IFQL** (Park et al., 2025) combines implicit value learning with a behavior flow and selects the highest-Q action among multiple flow samples at test time (Hansen-Estruch et al., 2023). **FQL** (Park et al., 2025) trains a one-step noise-conditioned actor by maximizing critic values while regularizing it toward actions generated by a

learned behavior flow. **CEP** (Lu et al., 2023), adapted here to a frozen behavior flow, learns an energy model through contrastive prediction and applies its input gradients during action generation. **Q-Flow** (Doo et al., 2026) learns intermediate values from terminal Q-values obtained by continuing the actor from data-path anchors and uses their gradients to supervise flow-actor training. **QAM** (Li & Levine, 2026) constructs stepwise actor supervision from terminal critic gradients using lean adjoints computed with the behavior model's Jacobian along actor rollouts. **QAM-E** (Li & Levine, 2026) augments QAM with a separately learned edit policy (Dong et al., 2026) that makes bounded corrections to generated actions.

Table 6: Hyperparameters for the standard and advanced OGBench settings. Critic aggregation, ρ, and Fourier embedding size apply to Q-Flow and SVFM. Value and Soft-value denote outer Q and inner V, respectively.

Hyperparameter	**Standard** (Park et al., 2025)	**Advanced** (Li & Levine, 2026)
Critic ensemble	2	10
Fourier embedding size	16	64
Critic aggregation	Mean (default); Min (antmaze-giant, puzzle)	$\text{mean}_h - \rho\ \text{std}_h$
Pessimism ρ	—	0.5 (default); 0 (humanoidmaze)
Action chunk size	1	1 (default); 5 (scene, cube, puzzle)
Discount γ	0.99 (default); 0.995 (antmaze-giant, humanoidmaze, antsoccer)	0.99 (default); 0.995 (antmaze-giant, humanoidmaze)
Seeds	8	12
Gradient steps	1M	1M
Batch size	256	256
SVFM network and optimizer settings		
Policy width / depth	512 / 4 hidden layers	512 / 4 hidden layers
Value width / depth	512 / 4 hidden layers	512 / 4 hidden layers
Soft-value width / depth	512 / 4 hidden layers	512 / 4 hidden layers
Activation (all)	GELU	GELU
LayerNorm	Value, Soft-value	Value, Soft-value
Learning rate	3×10^{-4} (constant)	3×10^{-4} (constant)
Optimizer	Adam (Kingma & Ba, 2015)	Adam
Adam $(\beta_1, \beta_2, \epsilon)$	$(0.9, 0.999, 10^{-8})$	$(0.9, 0.999, 10^{-8})$

Q-Flow and SVFM. Both methods follow the critic aggregation and Fourier time embedding settings of Doo et al. (2026) in Table 6; other baselines retain their method-specific aggregation rules. In the advanced setting, manipulation tasks use a chunked critic $Q(s_t, a_{t:t+h})$((Li et al., 2025)). The advanced setting's pessimistic aggregation((Fang et al., 2025; Ghasemipour et al., 2022)) is applied wherever the critic ensemble is aggregated. For SVFM, this includes the terminal Q-values used in soft-value estimation and the regression target of the time-conditioned critic, as well as the TD target.

The adaptive-temperature rule and actor coefficient follow equation 11 and equation 10; the sampling and estimation procedure is given in Appendix E. The toy, real-world, and offline-to-online experiments specify their inherited settings and experiment-specific changes in Appendices I.1, J.1, and K.

FAC protocol. The comparison in Table 20 follows Chae et al. (2026): discount 0.995 on every domain, its per-domain critic settings (penalty weight, density threshold scheme, and log-density estimator), and minimum Q aggregation on the two antmaze domains and mean aggregation else-

Table 7: Fixed SVFM settings for the standard, advanced, and FAC OGBench experiments. The domain-specific (κ, g) values are selected separately as described in Appendix G.

Setting	Value
Continuations per anchor	$K = 8$
Minimum estimation time	$t_{\min} = 0.1$
Soft-value and actor minibatch	256 anchors
Floor on s_Q	10^{-3}
Guidance cap	Twice the mean behavior velocity
Behavior flow	Pretrained for 250 epochs, cached per (domain, seed), and frozen during offline training

where. All three methods share these protocol settings, and the two SVFM variants share one frozen behavior flow per (domain, seed). FAC uses its released implementation and published hyperparameters unchanged; the two SVFM variants are tuned independently as described in Appendix M.1.

G Hyperparameter Selection

This appendix describes hyperparameter selection for the standard and advanced settings and offline-to-online learning, including what was searched, how configurations are selected, how many evaluation seeds are reported, and which baseline entries are published values rather than our own runs. The evaluation regimes are defined in Appendix F.

Comparison protocol. The baseline studies select hyperparameters separately for each environment: for example, Doo et al. (2026) select a guidance coefficient, a discount, and a Q-aggregation rule per domain, while Li & Levine (2026) select an inverse temperature and an edit scale per domain. We use the authors' reported per-domain baseline configurations and likewise select SVFM's (κ, g) separately for each environment under the corresponding evaluation setting. The common (κ, g) search grid for the standard, advanced, and offline-to-online comparisons is $\kappa \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and $g \in \{0.25, 0.5\}$. SVFM's adaptive temperature is computed as $\lambda = \kappa^2 s_Q / g$ by equation 11; its fixed settings and protocol-specific configurations are listed in Appendix F and are not part of this grid. The seeds used for SVFM hyperparameter selection are disjoint from those used to compute the reported results. Evaluation seeds for each comparison are listed below.

G.1 Standard Setting

Soft-Value Flow Matching. The selected configuration (Table 9) is evaluated using 8 seeds, $\{10001, 20002, \ldots, 80008\}$, for the main offline RL comparison.

Baselines. IQL, ReBRAC, IDQL, CAC, FBRAC, IFQL and FQL are the values reported over 8 seeds in Table 3 of Park et al. (2025), which we did not re-run. Q-Flow is our own re-run with the official implementation, over the same 8 seeds and the same scoring rule, at the per-domain guidance coefficient reported by Doo et al. (2026): 0.2 on both antmaze domains, 1 on both humanoidmaze domains, 0.5 on antsoccer, 5 on scene and cube-single, 2 on cube-double and 20 on both puzzle domains.

CEP. Lu et al. (2023) evaluate CEP only on D4RL (Fu et al., 2020), with a fixed inverse temperature $\beta = 3$ and a guidance scale s tuned per task at evaluation, and report no results on OGBench. We therefore implement CEP on the same frozen behavior flow as SVFM and tune it on OGBench ourselves. The energy network is trained with the contrastive objective of CEP on $K = 32$ behavior actions per state, the critic uses their in-support backup with $\beta_Q = \beta$, and the guidance is applied for $t \geq 0.1$ with 10 Euler steps. The original CEP fixes $\beta = 3$ on every task. For a fair comparison with SVFM, whose (κ, g) is selected per domain, we additionally tune $\beta \in \{0.5, 1, 3, 5\}$, which contains their value, together with $s \in \{1, 2, 3, 5, 10\}$ for each domain. The selected configurations (Table 8) are evaluated using 8 seeds, $\{10001, 20002, \ldots, 80008\}$, on all 5 tasks.

Table 8: Selected (β, s) of CEP in the standard setting.

Environment	β	s
antmaze-large	0.5	3
antmaze-giant	1	2
humanoidmaze-medium	0.5	2
humanoidmaze-large	1	1
antsoccer	0.5	3
scene	0.5	2
puzzle-3x3	0.5	1
puzzle-4x4	0.5	2
cube-single	0.5	1
cube-double	0.5	2

Adaptive-temperature comparison. The adaptive-versus-constant-temperature comparison in Table 3 evaluates each domain's default task at 1M steps using three seeds, $\{10001, 20002, 30003\}$.

G.2 Advanced Setting

Soft-Value Flow Matching. The selected configuration (Table 10) is evaluated using 12 seeds, $\{10001, 20002, \ldots, 120012\}$, for the main offline RL comparison.

Baselines. ReBRAC, DAC, QSM, FBRAC, IFQL and FQL are the published values of Li & Levine (2026). QAM, QAM-E and Q-Flow are our own re-runs over 12 seeds under our scoring rule, at their authors' published per-domain values: QAM and QAM-E at the inverse temperature and edit scale of the released reproduction script, Q-Flow at the advanced-setting guidance coefficient of Doo et al. (2026).

Table 9: Selected (κ, g) for Soft-Value Flow Matching in the standard setting, Table 1.

Environment	κ	g
antmaze-large	0.8	0.25
antmaze-giant	0.4	0.5
humanoidmaze-medium	0.6	0.25
humanoidmaze-large	0.8	0.25
antsoccer	0.6	0.25
scene	0.6	0.5
puzzle-3x3	0.4	0.25
puzzle-4x4	0.4	0.5
cube-single	0.6	0.25
cube-double	0.4	0.25

Table 10: Selected (κ, g) for Soft-Value Flow Matching in the advanced setting, Table 2.

Environment	κ	g
antmaze-large	1.0	0.5
antmaze-giant	1.0	0.5
humanoidmaze-medium	0.6	0.25
humanoidmaze-large	0.8	0.25
scene-sparse	0.4	0.25
puzzle-3x3-sparse	0.2	0.25
puzzle-4x4-sparse	0.2	0.25
cube-double	0.4	0.5
cube-triple	0.6	0.5
cube-quadruple	0.2	0.25

G.3 Offline-to-Online Learning

The baselines follow the hyperparameters of Doo et al. (2026) and Park et al. (2025).

We select SVFM's online-phase configuration separately from its offline configuration for each environment (Table 11), including the online base-flow option described below. The selected configurations are evaluated using 8 seeds, $\{10001, 20002, \ldots, 80008\}$.

The online base-flow options are the behavior flow, kept frozen during offline training, unfrozen at the start of the online phase and trained with the flow-matching loss on the replay buffer, a copy of the policy refreshed every 500K online steps, or the current policy itself. In the online phase, the actions of the policy are added to the replay buffer, so the critic is trained on the actions the policy takes. The over-estimation of unseen actions that motivates the fixed behavior continuations offline is therefore corrected by online data, and drawing the continuations from the current policy or from its periodically refreshed copy is a natural choice for this phase. With these choices, SVFM also performs well in the online phase (Figure 2).

Table 11: Per-domain online-phase hyperparameters of SVFM in the offline-to-online experiment under the protocol of Doo et al. (2026).

Environment	κ	g	online base
antmaze-giant	0.6	0.25	copy
humanoidmaze-medium	0.6	0.25	copy
antsoccer-arena	0.6	0.25	policy
cube-double	0.4	0.25	policy
puzzle-4x4	0.4	0.5	copy

H PER-TASK RESULTS

Tables 12 and 13 list the per-task results behind Tables 1 and 2 for every OGBench domain and each of its 5 tasks.

Table 12: Per-task success rates (%) under the standard setting of Park et al. (2025), for every method of Table 1; each entry is mean ± standard deviation over 8 seeds, and the mean over the 5 tasks is the entry of Table 1. (∗) denotes the default task per domain. Methods marked with † were trained by us: Q-Flow with the official implementation and the hyper-parameters reported in Doo et al. (2026), CEP at the hyper-parameters selected in Appendix G.1, and SVFM at its selected (κ, g), all scored as the mean success rate of the last 3 evaluations (800K, 900K, 1M steps). The remaining columns are taken from the FQL paper.

	Gaussian		Diffusion		Flow					
Domain / task	IQL	ReBRAC	IDQL	CAC	FBRAC	IFQL	FQL	CEP†	Q-Flow†	SVFM (ours)†
antmaze-large task 1 (∗)	48.0 ±9.0	91.0 ±10.0	0.0 ±0.0	42.0 ±7.0	70.0 ±20.0	24.0 ±17.0	80.0 ±8.0	50.8 ±10.0	96.1 ±1.4	90.8 ±2.1
antmaze-large task 2	42.0 ±6.0	88.0 ±4.0	14.0 ±8.0	1.0 ±1.0	35.0 ±12.0	8.0 ±3.0	57.0 ±10.0	41.6 ±6.8	85.3 ±2.5	86.2 ±4.2
antmaze-large task 3	72.0 ±7.0	51.0 ±18.0	26.0 ±8.0	49.0 ±10.0	83.0 ±15.0	52.0 ±17.0	93.0 ±3.0	80.6 ±7.2	93.9 ±2.4	94.3 ±1.9
antmaze-large task 4	51.0 ±9.0	84.0 ±7.0	62.0 ±25.0	17.0 ±6.0	37.0 ±18.0	18.0 ±8.0	80.0 ±4.0	70.2 ±5.4	68.8 ±25.0	89.2 ±2.6
antmaze-large task 5	54.0 ±22.0	90.0 ±2.0	2.0 ±2.0	55.0 ±6.0	76.0 ±8.0	38.0 ±18.0	83.0 ±4.0	74.1 ±5.1	91.5 ±2.0	89.0 ±2.4
antmaze-giant task 1 (∗)	0.0 ±0.0	27.0 ±22.0	0.0 ±0.0	0.0 ±0.0	0.0 ±1.0	0.0 ±0.0	4.0 ±5.0	0.0 ±0.0	13.6 ±10.1	1.8 ±3.0
antmaze-giant task 2	1.0 ±1.0	16.0 ±17.0	0.0 ±0.0	0.0 ±0.0	4.0 ±7.0	0.0 ±0.0	9.0 ±7.0	3.2 ±2.9	8.5 ±9.3	34.6 ±23.4
antmaze-giant task 3	0.0 ±0.0	34.0 ±22.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±1.0	0.2 ±0.3	19.5 ±17.5	2.0 ±3.3
antmaze-giant task 4	0.0 ±0.0	5.0 ±12.0	0.0 ±0.0	0.0 ±0.0	9.0 ±4.0	0.0 ±0.0	14.0 ±23.0	6.1 ±4.1	43.6 ±29.8	10.8 ±15.0
antmaze-giant task 5	19.0 ±7.0	49.0 ±22.0	0.0 ±1.0	0.0 ±0.0	6.0 ±10.0	13.0 ±9.0	16.0 ±28.0	29.8 ±11.5	76.1 ±9.7	54.8 ±22.1
humanoidmaze-medium task 1 (∗)	32.0 ±7.0	16.0 ±9.0	1.0 ±1.0	38.0 ±19.0	25.0 ±8.0	69.0 ±19.0	19.0 ±12.0	22.4 ±6.9	86.4 ±2.9	62.5 ±16.9
humanoidmaze-medium task 2	41.0 ±9.0	18.0 ±16.0	1.0 ±1.0	47.0 ±35.0	76.0 ±10.0	85.0 ±11.0	94.0 ±3.0	73.8 ±6.8	95.2 ±1.9	98.8 ±1.0
humanoidmaze-medium task 3	25.0 ±5.0	36.0 ±13.0	0.0 ±1.0	83.0 ±18.0	27.0 ±11.0	49.0 ±49.0	74.0 ±18.0	81.0 ±3.6	92.6 ±3.6	97.2 ±1.6
humanoidmaze-medium task 4	0.0 ±1.0	15.0 ±16.0	1.0 ±1.0	5.0 ±4.0	1.0 ±2.0	1.0 ±1.0	3.0 ±4.0	29.7 ±5.7	39.8 ±17.6	52.2 ±25.4
humanoidmaze-medium task 5	66.0 ±4.0	24.0 ±20.0	1.0 ±1.0	91.0 ±5.0	63.0 ±9.0	98.0 ±2.0	97.0 ±2.0	86.8 ±5.5	98.8 ±0.9	99.1 ±0.9
humanoidmaze-large task 1 (∗)	3.0 ±1.0	2.0 ±1.0	0.0 ±0.0	1.0 ±1.0	0.0 ±1.0	6.0 ±2.0	7.0 ±6.0	0.9 ±0.9	10.8 ±1.5	18.2 ±9.2
humanoidmaze-large task 2	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	1.2 ±0.8
humanoidmaze-large task 3	7.0 ±3.0	8.0 ±4.0	3.0 ±1.0	2.0 ±3.0	10.0 ±2.0	48.0 ±10.0	11.0 ±7.0	9.2 ±1.8	16.2 ±4.0	32.6 ±7.2
humanoidmaze-large task 4	1.0 ±0.0	1.0 ±1.0	0.0 ±0.0	0.0 ±1.0	0.0 ±0.0	1.0 ±1.0	2.0 ±3.0	4.7 ±0.9	3.0 ±2.7	29.6 ±10.6
humanoidmaze-large task 5	1.0 ±1.0	2.0 ±2.0	0.0 ±0.0	0.0 ±0.0	1.0 ±1.0	0.0 ±0.0	1.0 ±3.0	2.6 ±1.2	4.3 ±4.4	19.3 ±12.3
antsoccer task 1	14.0 ±5.0	0.0 ±0.0	44.0 ±12.0	1.0 ±3.0	17.0 ±3.0	61.0 ±25.0	77.0 ±4.0	88.9 ±4.0	75.1 ±4.5	58.2 ±5.5
antsoccer task 2	17.0 ±7.0	0.0 ±1.0	15.0 ±12.0	0.0 ±0.0	8.0 ±2.0	75.0 ±3.0	88.0 ±3.0	88.6 ±3.6	94.3 ±2.5	80.6 ±5.0
antsoccer task 3	6.0 ±4.0	0.0 ±0.0	0.0 ±0.0	8.0 ±19.0	16.0 ±3.0	14.0 ±22.0	61.0 ±6.0	57.6 ±7.1	52.9 ±4.4	62.3 ±5.3
antsoccer task 4 (∗)	3.0 ±2.0	0.0 ±0.0	0.0 ±1.0	0.0 ±0.0	24.0 ±4.0	16.0 ±9.0	39.0 ±6.0	52.5 ±4.1	28.3 ±5.1	35.3 ±5.5
antsoccer task 5	2.0 ±2.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	15.0 ±4.0	0.0 ±1.0	36.0 ±9.0	38.5 ±8.0	24.2 ±9.6	51.3 ±8.0
scene task 1	94.0 ±3.0	95.0 ±2.0	100.0 ±0.0	100.0 ±1.0	96.0 ±8.0	98.0 ±3.0	100.0 ±0.0	48.2 ±26.3	99.9 ±0.2	100.0 ±0.0
scene task 2 (∗)	12.0 ±3.0	50.0 ±13.0	33.0 ±14.0	50.0 ±40.0	46.0 ±10.0	0.0 ±0.0	76.0 ±9.0	93.5 ±5.7	97.4 ±2.4	97.2 ±3.3
scene task 3	32.0 ±7.0	55.0 ±16.0	94.0 ±4.0	49.0 ±16.0	78.0 ±14.0	54.0 ±19.0	98.0 ±1.0	31.8 ±26.7	97.4 ±1.7	98.0 ±1.1
scene task 4	0.0 ±1.0	3.0 ±3.0	4.0 ±3.0	0.0 ±0.0	4.0 ±4.0	0.0 ±0.0	5.0 ±1.0	0.3 ±0.7	7.9 ±5.3	4.3 ±6.1
scene task 5	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
puzzle-3x3 task 1	33.0 ±6.0	97.0 ±4.0	52.0 ±12.0	97.0 ±2.0	63.0 ±19.0	94.0 ±3.0	90.0 ±4.0	89.8 ±3.5	94.8 ±1.7	97.3 ±1.2
puzzle-3x3 task 2	4.0 ±3.0	1.0 ±1.0	0.0 ±1.0	0.0 ±0.0	2.0 ±2.0	1.0 ±2.0	16.0 ±5.0	9.6 ±5.9	45.5 ±7.3	84.0 ±4.1
puzzle-3x3 task 3	3.0 ±2.0	3.0 ±1.0	0.0 ±0.0	0.0 ±0.0	1.0 ±1.0	0.0 ±0.0	10.0 ±3.0	3.7 ±1.8	27.0 ±6.5	50.0 ±6.5
puzzle-3x3 task 4 (∗)	2.0 ±1.0	2.0 ±1.0	0.0 ±0.0	0.0 ±0.0	2.0 ±2.0	0.0 ±0.0	16.0 ±5.0	3.5 ±1.6	38.8 ±6.9	60.0 ±7.0
puzzle-3x3 task 5	3.0 ±2.0	5.0 ±3.0	0.0 ±0.0	0.0 ±0.0	2.0 ±2.0	0.0 ±0.0	16.0 ±3.0	16.3 ±11.4	34.2 ±3.2	48.4 ±6.4
puzzle-4x4 task 1	12.0 ±2.0	26.0 ±4.0	48.0 ±5.0	44.0 ±10.0	32.0 ±9.0	49.0 ±9.0	34.0 ±8.0	19.8 ±7.4	52.1 ±6.0	56.2 ±5.9
puzzle-4x4 task 2	7.0 ±4.0	12.0 ±4.0	14.0 ±5.0	0.0 ±0.0	5.0 ±3.0	4.0 ±4.0	16.0 ±5.0	0.4 ±0.5	18.9 ±2.8	23.3 ±2.3
puzzle-4x4 task 3	9.0 ±3.0	15.0 ±3.0	34.0 ±5.0	29.0 ±12.0	20.0 ±10.0	50.0 ±14.0	18.0 ±5.0	2.9 ±3.4	46.8 ±7.1	53.1 ±12.3
puzzle-4x4 task 4 (∗)	5.0 ±2.0	10.0 ±3.0	26.0 ±6.0	1.0 ±1.0	5.0 ±1.0	21.0 ±11.0	11.0 ±3.0	0.2 ±0.4	16.7 ±3.1	23.5 ±4.6
puzzle-4x4 task 5	4.0 ±1.0	7.0 ±3.0	24.0 ±11.0	0.0 ±0.0	4.0 ±3.0	2.0 ±2.0	7.0 ±3.0	0.4 ±0.5	11.2 ±2.9	11.2 ±1.9
cube-single task 1	88.0 ±3.0	89.0 ±5.0	95.0 ±2.0	77.0 ±28.0	73.0 ±33.0	79.0 ±4.0	97.0 ±2.0	96.4 ±2.1	95.2 ±1.4	93.6 ±2.5
cube-single task 2 (∗)	85.0 ±8.0	92.0 ±4.0	96.0 ±2.0	80.0 ±30.0	83.0 ±13.0	73.0 ±3.0	97.0 ±2.0	99.4 ±0.5	95.8 ±1.7	97.1 ±1.8
cube-single task 3	91.0 ±5.0	93.0 ±3.0	99.0 ±1.0	98.0 ±1.0	82.0 ±12.0	88.0 ±4.0	98.0 ±2.0	98.2 ±1.1	98.0 ±0.9	95.3 ±2.1
cube-single task 4	73.0 ±6.0	92.0 ±3.0	93.0 ±4.0	91.0 ±2.0	79.0 ±20.0	79.0 ±6.0	94.0 ±3.0	95.4 ±1.7	90.8 ±4.7	93.2 ±3.6
cube-single task 5	78.0 ±9.0	87.0 ±8.0	90.0 ±6.0	80.0 ±20.0	76.0 ±33.0	77.0 ±7.0	93.0 ±3.0	92.7 ±2.2	92.1 ±4.1	88.8 ±4.2
cube-double task 1	27.0 ±5.0	45.0 ±6.0	39.0 ±19.0	21.0 ±8.0	47.0 ±11.0	35.0 ±9.0	61.0 ±9.0	73.6 ±8.6	61.1 ±14.1	63.2 ±8.2
cube-double task 2 (∗)	1.0 ±1.0	7.0 ±3.0	16.0 ±10.0	2.0 ±2.0	22.0 ±12.0	9.0 ±5.0	36.0 ±6.0	65.2 ±4.8	30.4 ±5.5	42.1 ±12.1
cube-double task 3	0.0 ±0.0	4.0 ±1.0	17.0 ±8.0	3.0 ±1.0	4.0 ±2.0	8.0 ±5.0	22.0 ±5.0	62.8 ±9.8	25.8 ±5.9	47.6 ±6.7
cube-double task 4	0.0 ±0.0	1.0 ±1.0	0.0 ±1.0	0.0 ±1.0	0.0 ±1.0	1.0 ±1.0	5.0 ±2.0	9.8 ±1.8	6.0 ±1.7	12.7 ±5.3
cube-double task 5	4.0 ±3.0	4.0 ±2.0	1.0 ±1.0	3.0 ±2.0	2.0 ±2.0	17.0 ±6.0	19.0 ±10.0	74.0 ±11.0	35.3 ±10.6	46.8 ±12.6

Table 13: Per-task success rates (%) under the advanced setting of Li & Levine (2026), for every method of Table 2; each entry is mean ± standard deviation over 12 seeds, with approximate standard deviations for published baselines as described below, and the mean over the 5 tasks is the entry of Table 2; -sparse marks the sparse-reward variants. $(*)$ denotes the default task per domain. Methods marked with † were reproduced by us with the official hyper-parameters (QAM, QAM-E, Q-Flow) or run at the selected (κ, g) (SVFM), scored as the mean success rate of the last 3 evaluations. The remaining columns are taken from the QAM paper, which reports means with bootstrap 95% confidence intervals over 12 seeds; we approximate each standard deviation by multiplying the interval half-width by $\sqrt{12}/1.96$, using a normal-approximation heuristic rather than directly computed seed standard deviations.

	Gaussian	Diffusion		Flow						
Domain / task	ReBRAC	DAC	QSM	FBRAC	IFQL	FQL	QAM†	QAM-E†	Q-Flow†	SVFM (ours)†
antmaze-large task 1 $(*)$	98.0 ±1.8	88.0 ±8.0	87.0 ±13.3	0.0 ±0.0	41.0 ±23.9	93.0 ±6.2	79.1 ±8.7	84.3 ±6.3	96.3 ±1.3	90.4 ±2.3
antmaze-large task 2	88.0 ±5.3	71.0 ±7.1	78.0 ±12.4	0.0 ±0.0	14.0 ±8.0	86.0 ±5.3	75.6 ±11.7	75.6 ±2.7	87.4 ±2.3	86.2 ±3.0
antmaze-large task 3	98.0 ±1.8	98.0 ±1.8	99.0 ±1.8	12.0 ±13.3	54.0 ±16.8	59.0 ±17.7	86.8 ±3.6	92.7 ±2.3	96.8 ±1.2	95.6 ±1.5
antmaze-large task 4	94.0 ±2.7	91.0 ±5.3	91.0 ±5.3	0.0 ±0.0	25.0 ±8.0	54.0 ±23.9	69.8 ±12.6	58.1 ±15.8	91.8 ±1.3	87.3 ±3.1
antmaze-large task 5	95.0 ±1.8	92.0 ±3.5	96.0 ±2.7	0.0 ±0.0	45.0 ±19.4	86.0 ±4.4	86.9 ±3.3	83.8 ±3.5	93.3 ±2.0	88.8 ±2.6
antmaze-giant task 1 $(*)$	36.0 ±8.8	53.0 ±14.1	62.0 ±21.2	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	10.3 ±4.3	0.5 ±1.1	43.1 ±7.7	34.8 ±9.6
antmaze-giant task 2	73.0 ±8.8	0.0 ±0.0	1.0 ±1.8	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	5.1 ±4.3	46.8 ±5.6
antmaze-giant task 3	13.0 ±8.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	1.0 ±1.8	0.0 ±0.0	0.2 ±0.4	0.0 ±0.0	17.4 ±12.8	18.9 ±7.5
antmaze-giant task 4	74.0 ±22.1	0.0 ±0.0	0.0 ±0.9	0.0 ±0.0	2.0 ±2.7	0.0 ±0.0	13.6 ±11.9	0.0 ±0.0	66.1 ±20.2	56.9 ±16.0
antmaze-giant task 5	89.0 ±6.2	25.0 ±36.2	58.0 ±36.2	0.0 ±0.0	2.0 ±4.4	0.0 ±0.0	53.2 ±30.4	0.0 ±0.0	71.1 ±9.0	81.5 ±4.5
humanoidmaze-medium task 1 $(*)$	38.0 ±20.3	87.0 ±14.1	88.0 ±8.0	26.0 ±8.0	86.0 ±3.5	34.0 ±24.7	19.5 ±10.8	16.8 ±13.1	87.3 ±3.2	70.9 ±22.1
humanoidmaze-medium task 2	91.0 ±15.9	96.0 ±2.7	96.0 ±3.5	78.0 ±6.2	92.0 ±4.4	95.0 ±6.2	89.4 ±27.0	90.1 ±27.2	93.3 ±2.7	98.0 ±1.9
humanoidmaze-medium task 3	83.0 ±31.8	92.0 ±9.7	95.0 ±4.4	28.0 ±13.3	93.0 ±3.5	96.0 ±2.7	92.8 ±3.6	55.5 ±20.7	93.5 ±2.5	98.1 ±1.5
humanoidmaze-medium task 4	37.0 ±30.0	43.0 ±15.9	31.0 ±14.1	3.0 ±3.5	60.0 ±8.8	14.0 ±19.4	7.8 ±17.9	0.0 ±0.0	48.3 ±14.7	47.1 ±19.9
humanoidmaze-medium task 5	96.0 ±3.5	99.0 ±0.9	98.0 ±1.8	59.0 ±16.8	98.0 ±2.7	99.0 ±1.8	98.9 ±0.9	90.7 ±27.4	98.3 ±0.7	98.8 ±0.9
humanoidmaze-large task 1 $(*)$	36.0 ±12.4	0.0 ±0.9	10.0 ±4.4	0.0 ±0.0	36.0 ±8.8	8.0 ±5.3	2.4 ±1.9	5.7 ±3.5	10.9 ±4.0	21.0 ±6.5
humanoidmaze-large task 2	1.0 ±1.8	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.1 ±0.2	0.0 ±0.0	0.1 ±0.2	1.2 ±0.9
humanoidmaze-large task 3	32.0 ±15.9	0.0 ±0.0	16.0 ±7.1	0.0 ±0.9	55.0 ±8.0	19.0 ±7.1	14.7 ±10.7	5.2 ±1.8	13.5 ±5.1	29.7 ±7.4
humanoidmaze-large task 4	10.0 ±10.6	0.0 ±0.0	2.0 ±1.8	0.0 ±0.0	2.0 ±1.8	9.0 ±6.2	9.9 ±5.9	0.2 ±0.4	3.6 ±1.9	37.3 ±7.6
humanoidmaze-large task 5	7.0 ±8.0	1.0 ±1.8	2.0 ±3.5	0.0 ±0.0	29.0 ±23.9	9.0 ±7.1	10.4 ±9.3	0.0 ±0.0	4.1 ±2.4	34.6 ±10.6
scene-sparse task 1	98.0 ±2.7	100.0 ±0.9	100.0 ±0.0	66.0 ±20.3	93.0 ±3.5	99.0 ±0.9	100.0 ±0.0	100.0 ±0.0	99.6 ±0.8	100.0 ±0.0
scene-sparse task 2 $(*)$	90.0 ±5.3	99.0 ±1.8	78.0 ±7.1	80.0 ±12.4	63.0 ±9.7	71.0 ±9.7	98.8 ±0.8	99.1 ±1.3	99.8 ±0.4	100.0 ±0.0
scene-sparse task 3	51.0 ±15.0	79.0 ±8.8	97.0 ±1.8	41.0 ±35.3	71.0 ±8.0	97.0 ±1.8	99.6 ±0.6	99.7 ±0.5	97.9 ±1.4	99.1 ±0.7
scene-sparse task 4	60.0 ±20.3	8.0 ±11.5	92.0 ±4.4	49.0 ±40.7	98.0 ±1.8	92.0 ±3.5	99.4 ±0.8	99.4 ±0.7	98.7 ±1.0	97.9 ±4.2
scene-sparse task 5	27.0 ±28.3	53.0 ±18.6	25.0 ±5.3	17.0 ±21.2	95.0 ±3.5	33.0 ±7.1	89.2 ±2.6	88.6 ±2.8	90.8 ±5.1	92.0 ±7.1
puzzle-3x3-sparse task 1	99.0 ±1.8	98.0 ±3.5	89.0 ±24.7	1.0 ±1.8	100.0 ±0.0	99.0 ±1.8	99.3 ±0.9	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0
puzzle-3x3-sparse task 2	77.0 ±26.5	66.0 ±40.7	85.0 ±25.6	0.0 ±0.0	100.0 ±0.0	79.0 ±33.6	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0
puzzle-3x3-sparse task 3	85.0 ±7.1	54.0 ±32.7	18.0 ±26.5	0.0 ±0.0	100.0 ±0.0	83.0 ±37.1	99.3 ±1.3	100.0 ±0.0	99.9 ±0.2	100.0 ±0.0
puzzle-3x3-sparse task 4 $(*)$	62.0 ±28.3	72.0 ±10.6	87.0 ±6.2	0.0 ±0.0	100.0 ±0.0	84.0 ±33.6	99.9 ±0.2	99.9 ±0.2	100.0 ±0.0	99.9 ±0.2
puzzle-3x3-sparse task 5	70.0 ±32.7	51.0 ±40.7	9.0 ±15.0	1.0 ±1.8	100.0 ±0.0	6.0 ±6.2	99.9 ±0.2	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0
puzzle-4x4-sparse task 1	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	29.0 ±17.7	0.0 ±0.0	17.0 ±17.7	0.3 ±0.4	78.5 ±14.5	0.3 ±0.6	0.4 ±0.4
puzzle-4x4-sparse task 2	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	13.0 ±12.4	1.0 ±0.9	1.0 ±1.8	0.2 ±0.4	14.0 ±17.6	0.1 ±0.2	0.3 ±0.5
puzzle-4x4-sparse task 3	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	16.0 ±15.0	0.0 ±0.9	3.0 ±5.3	0.2 ±0.3	43.4 ±6.4	0.1 ±0.2	0.2 ±0.3
puzzle-4x4-sparse task 4 $(*)$	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	6.0 ±7.1	0.0 ±0.0	2.0 ±4.4	0.1 ±0.2	30.9 ±17.2	0.1 ±0.2	0.2 ±0.4
puzzle-4x4-sparse task 5	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	13.0 ±18.6	0.0 ±0.0	2.0 ±5.3	0.2 ±0.4	26.8 ±17.3	0.1 ±0.2	0.3 ±0.3
cube-double task 1	29.0 ±9.7	36.0 ±7.1	80.0 ±8.8	0.0 ±0.0	16.0 ±3.5	81.0 ±8.0	84.2 ±4.1	86.8 ±3.0	61.8 ±6.9	84.3 ±5.4
cube-double task 2 $(*)$	6.0 ±6.2	35.0 ±8.8	29.0 ±7.1	0.0 ±0.0	12.0 ±3.5	46.0 ±10.6	78.4 ±13.4	78.5 ±6.6	32.6 ±7.4	67.0 ±5.6
cube-double task 3	2.0 ±3.5	31.0 ±8.8	33.0 ±8.0	0.0 ±0.0	10.0 ±3.5	42.0 ±12.4	48.6 ±9.7	48.3 ±9.2	22.0 ±4.5	76.2 ±3.8
cube-double task 4	1.0 ±1.8	16.0 ±4.4	5.0 ±4.4	0.0 ±0.0	4.0 ±3.5	10.0 ±3.5	23.0 ±6.0	16.3 ±5.4	12.3 ±4.7	29.7 ±5.3
cube-double task 5	4.0 ±4.4	56.0 ±8.8	16.0 ±6.2	0.0 ±0.0	11.0 ±5.3	50.0 ±11.5	82.8 ±5.8	80.0 ±4.7	52.7 ±7.8	86.1 ±4.8
cube-triple task 1	4.0 ±2.7	24.0 ±12.4	26.0 ±8.8	2.0 ±2.7	2.0 ±0.9	15.0 ±8.0	11.4 ±4.8	18.8 ±5.5	11.1 ±3.8	8.2 ±3.0
cube-triple task 2 $(*)$	0.0 ±0.0	0.0 ±0.9	1.0 ±1.8	0.0 ±0.0	0.0 ±0.0	0.0 ±0.9	0.3 ±0.4	1.7 ±0.7	0.8 ±0.8	0.4 ±0.6
cube-triple task 3	0.0 ±0.0	0.0 ±0.0	0.0 ±0.9	0.0 ±0.0	0.0 ±0.0	1.0 ±0.9	2.2 ±1.2	3.7 ±1.8	1.8 ±1.2	1.1 ±1.1
cube-triple task 4	0.0 ±0.0	0.0 ±0.0	0.0 ±0.9	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.4 ±0.6	1.3 ±0.9	0.2 ±0.3	0.1 ±0.2
cube-triple task 5	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.1 ±0.2	2.8 ±2.3
cube-quadruple task 1	35.0 ±22.1	12.0 ±19.4	90.0 ±4.4	0.0 ±0.0	8.0 ±5.3	11.0 ±15.9	11.7 ±4.2	23.0 ±14.4	37.2 ±24.1	97.7 ±1.8
cube-quadruple task 2 $(*)$	6.0 ±9.7	1.0 ±0.9	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.1 ±0.4	13.4 ±13.1	76.7 ±9.6
cube-quadruple task 3	3.0 ±2.7	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	2.0 ±3.5	0.0 ±0.0	0.6 ±0.6	0.1 ±0.2	0.5 ±1.3	49.9 ±7.6
cube-quadruple task 4	0.0 ±0.0	0.0 ±0.0	4.0 ±5.3	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.3 ±0.6	24.2 ±4.7
cube-quadruple task 5	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0

I Critic Queries for Policy Supervision

This section measures where each method queries the critic to supervise its actor, first in a toy experiment with a known over-estimated OOD region (Appendix I.1) and then on OGBench (Appendix I.2).

I.1 Toy Experiment

This subsection describes the two-dimensional experiment behind Figure 1. It asks how strongly each policy-improvement rule amplifies an off-support over-estimation of the critic. All methods receive the same critic, including its error. The experiment therefore measures the robustness of the improvement step, not the accuracy of the critic.

Environment and data. The task is a stateless bandit with a two-dimensional action $a \in \mathbb{R}^2$. The behavior data consist of 100,000 actions drawn from an equal mixture of three isotropic Gaussians with standard deviation 0.35, centred on a circle of radius 2.4 at 90°, 210° and 330°. The rewards of the three modes are 0, 0.5 and 1, with additive label noise $\mathcal{N}(0, 0.05^2)$. The centroid of the modes, the origin, lies 6.9 standard deviations from every mode and has essentially no behavior density.

Shared critic and the OOD bump. For each seed we train one behavior flow and one critic. The behavior flow is a conditional flow-matching velocity field $u_{\mathrm{bc}}(x, t)$, and the critic $\hat{Q}(a)$ is regressed on the observed rewards with a squared loss. Both are ReLU MLPs with four hidden layers of width 256 and are trained jointly for 4,000 steps with Adam (learning rate 3×10^{-4}, batch 1,024). To emulate an over-estimation off the data support, we add a Gaussian bump at the origin,

$$Q_b(a) = \hat{Q}(a) + b\,\sigma_{\mathcal{D}} \exp\left(-\frac{\|a\|^2}{2 \cdot 0.6^2}\right), \qquad \sigma_{\mathcal{D}} = \mathrm{std}_{a\sim\mathcal{D}}\,\hat{Q}(a),$$

where $b = 6$ sets the bump height in units of the critic's spread over the data ($\sigma_{\mathcal{D}} \approx 0.41$). The learned critic already extrapolates to about 0.7 at the origin without the bump, so the bump exceeds the value of the best mode (about 1.0) on every seed. The true reward at the origin is zero. Within a seed, every method is given the same Q_b, with identical parameters.

Methods. Every method starts from the same behavior flow and runs 2,000 improvement updates with Adam (learning rate 3×10^{-4}, batch 1,024, 10 Euler steps for sampling). QAM, Q-Flow and SVFM use the same constant temperature $\lambda = 0.3$, so that they target the same tilted distribution, and FQL uses $\alpha = 1$. The methods differ only in where the critic is evaluated.

- **QAM** (Li & Levine, 2026): adjoint matching. The terminal condition of the adjoint is $-\lambda^{-1}\nabla_a Q_b$ at the endpoint of the current actor's controlled SDE. Following the original implementation, gradients are clipped to global norm 1.
- **Q-Flow** (Doo et al., 2026): an inner value network $V_\psi(x_t, t)$ (four hidden layers of width 128) is regressed on Q_b evaluated at the endpoint of the current actor's ODE flow from (x_t, t). The actor regresses onto $(x_1 - x_0) + \lambda^{-1}\nabla_x V_\psi(x_t, t)$.
- **FQL** (Park et al., 2025): a one-step policy $\pi(z)$ minimises $-Q_b(\pi(z))/\overline{|Q_b|} + \alpha\|\pi(z) - \Psi_{\mathrm{bc}}(z)\|^2$, where Ψ_{bc} is the noise-to-action map of the behavior flow. Because the critic gradient is taken at $\pi(z)$, FQL also queries the critic at the current actor's actions. The one-step policy is first distilled onto Ψ_{bc} for 2,000 updates without the critic, so that it starts from the behavior policy like the other methods.
- **SVFM (ours)**: the soft value at (x_t, t) is estimated from K continuations of the frozen behavior flow, and the critic is evaluated only at these behavior endpoints. In this experiment SVFM uses this constant temperature instead of the adaptive temperature. We reuse K and $t_{\min}$ from Table 7, with $\kappa = 0.6$ and the guidance clip at 2 times the behavior velocity.

Figure 1. The figure shows the critic queries of the last update (2,000). The queries of the 5 seeds are pooled ($5 \times 1{,}024$ per method), and 1,500 of them are drawn at random for display; the percentage in each panel is computed on all 5,120. The grey shading is the behavior density, the dashed circle, labelled *over-estimated OOD region*, marks the bump and is drawn at two bump widths (radius

1.2) around the origin, and the background shows no critic values. Every point outside the behavior support counts as OOD, not only the points inside this circle.

Metrics. Every 50 updates we record up to 1,024 actions at which the critic was evaluated in that update, together with 8,192 samples from the current policy. An action is *out of distribution* (OOD) if the behavior mixture density there is below 1% of the density at a mode centre, and it is *in the bump* if $\|a\| < 0.9$. We report the fraction of critic queries that are OOD, the fraction of policy mass inside the bump, and the true reward: the bump-free mode reward under the Gaussian responsibilities, scored as zero for OOD samples. The behavior policy itself scores a true reward of 0.48 with 3.5% of its mass OOD.

Results. Table 14 lists the values at the last update. 19.1% (FQL), 12.6% (QAM) and 39.3% (Q-Flow) of the critic queries are OOD, against 1.3% for SVFM, whose queries stay on the behavior support regardless of the critic's error.

Table 14: Toy experiment after 2,000 updates, mean over 5 seeds.

Method	OOD queries	Policy mass OOD	Policy mass in bump	True reward
FQL	19.1%	17.8%	16.7%	0.43
QAM	12.6%	6.9%	3.2%	0.60
Q-Flow	39.3%	84.1%	69.3%	0.13
SVFM (ours)	**1.3%**	**3.5%**	**0.4%**	**0.72**

I.2 OGBench

This subsection tests whether the effect in Figure 1 also appears on OGBench. We use the checkpoints after 1M offline steps of SVFM, Q-Flow, FQL and FBRAC in the *standard* setting of Table 1, on the default task of each of the 10 domains, with 3 seeds (10001, 20002, 30003).

Critic queries. For each checkpoint we take 2,048 dataset states. At each state s we collect the actions at which the actor-supervision step of the method evaluates the critic: the endpoints of the K behavior continuations for SVFM, the endpoint of the current actor's flow for Q-Flow, the one-step policy action for FQL, and the final action of the actor's flow for FBRAC. We compare them with 32 candidate actions sampled at s from the pretrained behavior flow. A state is counted as *above-support* when the mean critic value at the queries exceeds the critic value of every candidate, that is, when the critic ranks the queried actions above all behavior-like actions at s.

Results. Table 15 shows how often each method queries the critic at above-support states, and Figure 5 shows the queries at two such states. SVFM does so at 1.6% of the states, against 11.9% to 16.6% for FBRAC, Q-Flow and FQL, and it is the lowest in every domain. When SVFM draws its continuations from the current actor instead of the behavior flow, the fraction rises to 5.6% (Appendix M.2).

J Real-World Experiment Details

J.1 Experimental Setup

Hardware platform. The robot is a Universal Robots UR7e six-degree-of-freedom arm with a Robotiq 2F-85 parallel-jaw gripper. Two Intel RealSense D435 cameras provide RGB observations at 1280×720 and 30 fps: one fixed camera gives a third-person view of the workspace and one wrist camera is mounted on the gripper; depth is not used. Policies output joint-position targets at 15 Hz, which are smoothed and streamed as 250 Hz position commands to the robot driver through ROS 2. Policies are served from a single NVIDIA GeForce RTX 5070 Ti GPU during evaluation.

Tasks and data. Figure 6 shows the initial configurations and the required object motion of the 4 tasks: *carrot in pot* (place the carrot inside the pot), *bowl stacking* (place the orange bowl inside the purple bowl), *triple bowl stacking* (place both orange bowls into the purple bowl one after the other),

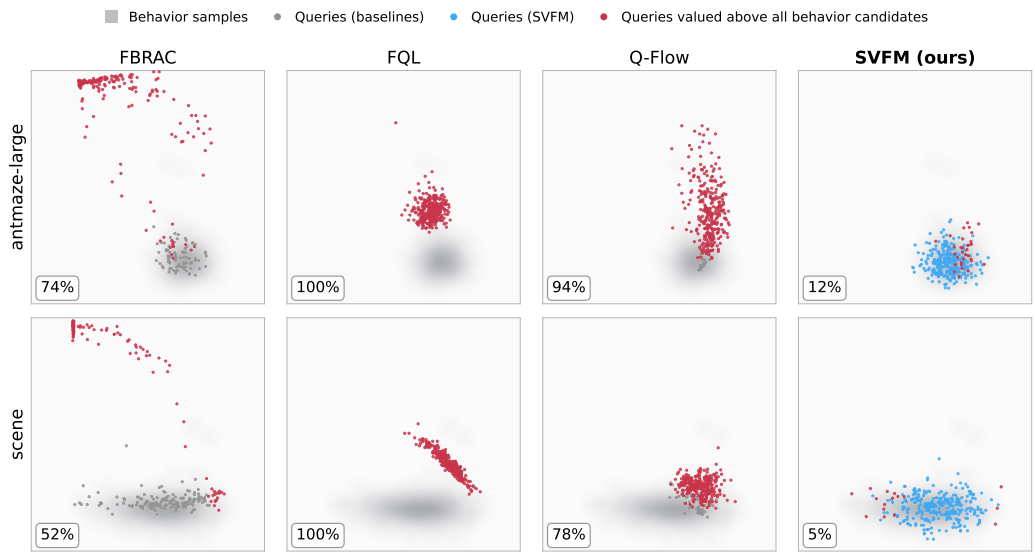


Figure 5: Critic queries of the actor-supervision step at one dataset state of antmaze-large and of scene, chosen among the states at which the queries of FBRAC, FQL and Q-Flow are all valued above every behavior candidate. The axes are the first two principal components of 2,000 samples of the behavior flow at that state. The percentage is the fraction of each method's queries that its critic values above all 32 behavior candidates.

Table 15: Fraction (%) of states at which the actor-supervision queries are ranked by the critic above all 32 behavior candidates, mean over 3 seeds.

Environment	FBRAC	FQL	Q-Flow	SVFM (ours)
antmaze-large	18.0	12.5	13.4	0.9
antmaze-giant	8.7	15.4	13.5	1.4
humanoidmaze-medium	7.9	32.9	8.2	1.3
humanoidmaze-large	5.8	19.4	6.4	1.2
antsoccer	6.3	20.0	15.3	1.4
scene	17.5	18.3	27.0	2.1
puzzle-3x3	30.6	14.5	12.1	2.2
puzzle-4x4	4.5	13.6	12.3	2.1
cube-single	14.0	12.2	20.3	2.1
cube-double	5.8	7.6	16.9	1.7
Average	11.9	16.6	14.5	1.6

and *cube stacking* (place the red cube on top of the blue cube). Demonstrations were collected by teleoperation with a GELLO leader arm (Wu et al., 2024) under fixed lighting at 30 Hz and subsampled to 15 Hz for training. A separate flow policy is trained for each task on its 50 demonstrations, and the evaluation configurations are taken from demonstrations outside this training set.

Implementation details. Across all real-world experiments, each action specifies six absolute joint-position targets and a gripper command. For the task-specific flow policies, we apply spatial softmax to ResNet-18 image features, concatenate them with proprioception, and feed the resulting representation to MLP policy and value networks. We pretrain BC for 250 epochs and use action chunks of $H = 8$, with actions normalized to $[-1, 1]$. The reward is -1 per policy step and 0 at termination, with discount $\gamma = 0.99$ per policy step.

Flow BC deploys the pretrained flow directly. BoN selects the highest-value chunk among 32 samples from Flow BC, using an IQL critic (Kostrikov et al., 2022) with two Q-networks, minimum aggregation, and expectile 0.9. Q-Flow (Doo et al., 2026) and SVFM train actors of the same ar-

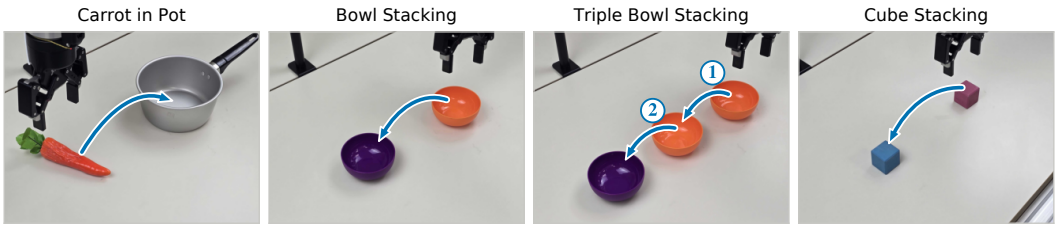


Figure 6: Real-world tasks. Initial configurations of the 4 manipulation tasks; arrows indicate the required object motion, and numbers give the order of the two placements in triple bowl stacking.

chitecture for 100K updates, using $\lambda = 5$ for Q-Flow and $(\kappa, g) = (0.6, 0.25)$ for SVFM. SVFM retains the frozen behavior flow for continuations and uses equation 9. Other network and optimizer settings, including Fourier time embeddings, follow the standard-setting cube-single configuration in Table 6, with the fixed SVFM settings in Table 7. We evaluate the final checkpoints without robot-specific hyperparameter tuning.

Evaluation protocol. Each method is trained once and evaluated over 20 trials per task, using the same robot starting pose and one or two fixed object configurations per task across methods. All methods execute complete chunks before replanning; only BoN uses candidate selection. Success requires full task completion and object release, including both placements for triple bowl stacking. We report the mean success rate over the 4 tasks.

SVFM on a pretrained VLA policy. We fine-tune GR00T N1.7 (Bjorck et al., 2025) jointly on all 200 demonstrations from the 4 tasks, using language instructions to identify the task. The GR00T policies use action chunks of $H = 16$. Training proceeds in three stages: BC fine-tuning, IQL critic learning, and SVFM adaptation. We first train BC for 10K updates with batch size 32, then fit an MLP IQL critic (Kostrikov et al., 2022) on the same demonstrations for 50K updates with batch size 64, expectile 0.8, and ten Q-networks. The critic's state input concatenates masked mean-pooled GR00T vision-language tokens, proprioception, and an embodiment indicator. SVFM starts from BC and trains LoRA adapters (Hu et al., 2022) on its action head together with the soft-value networks for 10K updates with batch size 32, keeping the original BC weights and IQL critic frozen. BC+ is trained independently from the same released model for 20K updates with batch size 32, so that its actor-update budget matches that of SVFM, which additionally trains the critic. BoN samples 8 action chunks from BC at every replanning step and executes the one with the highest value under the same frozen IQL critic (minimum over the ten Q-networks); it uses no additional training. We evaluate BC, BC+, BoN, and SVFM over 20 trials per task from the same fixed initial configurations, using the success criteria above; Table 16 reports all four policies, including the BC checkpoint that SVFM starts from, and Table 5 in the main text shows BC+, BoN, and SVFM.

J.2 Beyond Success Rate

Success rates do not show how much SVFM changes the behavior of the flow policy it starts from. Figure 4 shows 4 successful trials per method from the same initial object placement, while Table 17 compares four trajectory statistics over all 20 trials per task, computed from the measured joint states (forward kinematics to the tool center point) and the commanded gripper signal. Figure 7 shows, in the robot base frame, all successful Flow BC trials at that placement together with an equal number of the earliest successful SVFM trials, with markers at the start, the grasp (the last gripper closure before the task-completing release), and the release.

Trajectory statistics. Relative to Flow BC, SVFM travels 1.3–3.5× less distance per trial, switches the gripper 3–9× less often, and shows 1.6–2.7× less high-frequency joint-velocity jitter on every task. Between the first motion and the grasp, Flow BC meanders (straightness 0.35–0.62), whereas SVFM approaches the object as directly as the human demonstrations on 3 of the 4 tasks (0.76–0.83 vs. 0.79–0.85); on cube stacking both policies approach indirectly. Among successful

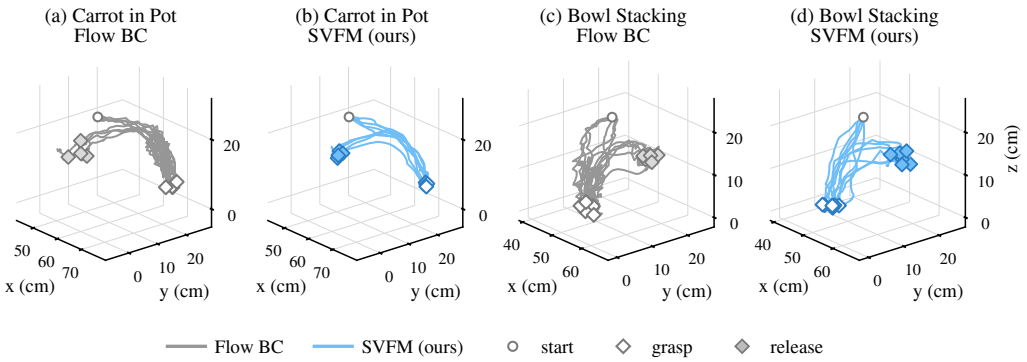


Figure 7: Three-dimensional view (robot base frame) of successful trajectories from the same initial object placements as Figure 4: all successful Flow BC trials of each task and an equal number of the earliest successful SVFM trials.

trials, SVFM also completes the task about twice as fast: it releases the carrot after 8.3 s and the bowl after 6.7 s (median), versus 19.7 s and 11.7 s for Flow BC (4 and 8 successful trials).

Where the smoother motion comes from. The improvement is already present in the action chunks themselves: the median second difference between consecutive targets within a chunk is 0.066–0.069 rad for Flow BC and 0.023–0.034 rad for SVFM (0.005 rad in the demonstrations), so the smoother motion originates in the policy output and not in execution, which is identical for all methods.

GR00T N1.7. The pretrained VLA policy shows the same kind of improvement over BC+, through a different mechanism: with SVFM, the action chunks of GR00T are not smoother inside a chunk (median step between consecutive targets 0.013–0.015 rad for both policies), but the jump at each replanning boundary halves, from 0.050–0.061 rad for BC+ to 0.022–0.032 rad for SVFM, so consecutive chunks join into one motion. Over successful trials, the SVFM end effector also travels less than BC, BC+, and BoN on every task except triple bowl stacking (Table 16).

Table 16: GR00T N1.7 real-world success rate (%, 20 trials per task) and median end-effector path length (m) over successful trials for all four policies. BC is the 10K-update checkpoint from which SVFM is fine-tuned. †: a single successful trial.

	BC		BC+		BoN		SVFM (ours)	
Task	SR (%)	Path (m)	SR (%)	Path (m)	SR (%)	Path (m)	SR (%)	Path (m)
Carrot in pot	60	1.57	**95**	1.21	65	1.54	**95**	**0.89**
Bowl stacking	95	0.65	**100**	0.61	**100**	0.60	**100**	**0.57**
Triple bowl stacking	65	1.44	90	1.44	75	**1.28**	**100**	1.36
Cube stacking	5	0.85†	5	1.65†	5	0.86†	**100**	**0.78**
Average	56	1.13	73	1.23	61	1.07	**99**	**0.90**

Cube placement. Cube stacking is the task on which the GR00T policies differ most (1/20 for both BC and BC+ versus 20/20 for SVFM), and Figure 8 shows where BC+ and SVFM release the cube: for every trial we take the grasp (a commanded gripper closure held for at least one second with the measured opening at the width of the cube) and plot the displacement of the cube from grasp to release, relative to the median displacement of the 19 trials that stacked the cube on the first placement. BC+ grasps the cube in 19 of 20 trials and carries it to the target. It then releases the cube a median of 27 mm from this center, always in the same direction along the transport motion (17 of 19 beyond the range at which stacking succeeded), so the cube slides off. The SVFM placements lie a median of 8 mm from the center, 13 of 20 within 10 mm, and 19 of 20 trials stack the cube on the first

Table 17: Trajectory statistics of the flow policies on the real robot: median over the 20 trials per task and method. Path length: end-effector travel per trial. Gripper toggles: open/close switches of the commanded gripper per trial. Jitter: RMS of the measured joint velocity above 4 Hz. Straightness: straight-line distance divided by path length between the first motion and the grasp (1 = straight); the value for the human demonstrations (Demo) is shown for reference.

	Path length (m)		Gripper toggles		Jitter (rad/s)		Approach straightness		
Task	Flow BC	SVFM	Flow BC	SVFM	Flow BC	SVFM	Demo	Flow BC	SVFM
Carrot in pot	2.66	**0.81**	12.0	**2.0**	0.032	**0.012**	0.79	0.62	**0.81**
Bowl stacking	2.14	**0.61**	18.0	**2.0**	0.030	**0.015**	0.85	0.58	**0.83**
Triple bowl stacking	2.50	**1.51**	13.0	**4.0**	0.031	**0.016**	0.81	0.35	**0.76**
Cube stacking	2.41	**1.92**	27.5	**5.5**	0.030	**0.019**	0.84	0.50	0.49

placement. This is consistent with value guidance moving the policy toward the terminal states of the demonstrations, in which the cube rests on the target, rather than toward the average demonstrated action; the demonstrations move the cube by the same amount as the successful placements to within 4 mm, so the offset is not inherited from the data.

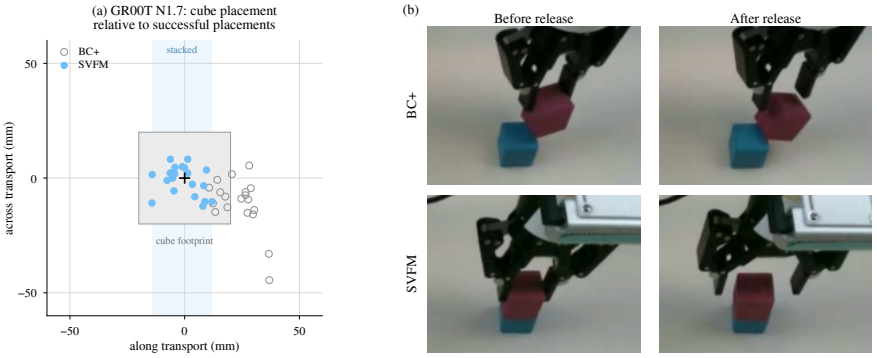


Figure 8: GR00T N1.7 (BC+) cube placement. (a) Grasp-to-release displacement on the first placement of each trial, centered at the median of 19 first-placement successes. The square shows the 40 mm cube footprint; the band marks the observed successful range along the transport direction. (b) Example BC+ (top) and SVFM (bottom) trials before and after release.

K Details of the Offline-to-Online Experiment

Figure 2 follows the offline-to-online protocol of Doo et al. (2026), on the 5 domains of their offline-to-online experiment with the default task of each: task 1 of antmaze-giant and humanoidmaze-medium, task 4 of antsoccer-arena, task 2 of cube-double and task 4 of puzzle-4x4. Every method aggregates its two critics by the mean. The discount is 0.995 on antmaze-giant, humanoidmaze-medium and antsoccer-arena and 0.99 on cube-double and puzzle-4x4. Each run trains for 1M gradient steps offline and then interacts with the environment for 1M steps, with one gradient update per environment step. The online transitions are added to a replay buffer that initially holds the offline data, and batches are drawn uniformly from it. The success rate is evaluated every 100K steps with 50 episodes, over 8 seeds.

Offline-to-online hyperparameter selection, online base-flow options, and selected configurations are detailed in Appendix G.3.

L Relation to CEP and QGPO

Shared guidance principle. SVFM shares the conditional log-partition construction used by Contrastive Energy Prediction (CEP) and its offline-RL application QGPO (Lu et al., 2023). QGPO defines an intermediate energy as

$$\mathcal{E}_\tau(s,z) = \log \mathbb{E}_{\mu(a|z,s)}\left[e^{\beta Q(s,a)}\right],$$

where $\mu(a \mid z, s)$ is the behavior posterior induced by its Gaussian noising kernel (Lu et al., 2023, Eq. (17)). With the same behavior distribution and critic, matched noising kernels, $\kappa = 1, \tau = 1-t$, and $\beta = 1/\lambda$, this quantity equals $V_{\lambda,1}(s,z,t)/\lambda$ (Appendix C). Thus, the conditional log-partition guidance is a shared foundation; the differences below concern its estimation and use in policy training.

Conditional continuations and regression targets. QGPO pre-generates behavior actions for each state, independently noises them into different intermediate actions, and trains an energy model by matching contrastive predictions to normalized exponential-Q labels (Lu et al., 2023, Eqs. (19)–(20)). SVFM fixes one intermediate anchor (s, x_t, t) and samples K terminal continuations conditional on that anchor. It regresses V_ψ onto their $\lambda \log \frac{1}{K}\sum_j e^{Q(s,a^{(j)})/\lambda}$. These K continuations estimate one conditional expectation, whereas QGPO contrasts different noisy actions. Both methods evaluate behavior-generated actions. CEP also considers an exponential regression alternative (Lu et al., 2023, Appendix H); SVFM's distinction is the conditional multi-continuation log-mean-exp target, rather than regression alone. Their finite-sample approximations differ: practical QGPO uses self-normalized labels, while SVFM's target has the Jensen bias analyzed in Appendix A.3. This comparison does not establish a bias or variance advantage for either estimator.

Using guidance to train an actor. QGPO applies the learned intermediate-energy gradient during policy sampling (Lu et al., 2023, Eq. (18)). SVFM uses the soft-value gradient to train a separate flow actor through equation 9. At deployment, integrating this actor requires no soft-value evaluations or input gradients. This changes where the guidance computation occurs; it does not by itself establish lower total training cost or faster inference.

Controlling the continuation law. SVFM's κ in equation 10 changes the conditional endpoint law $P_\kappa(X_1 \mid X_t = x)$ while preserving the behavior marginals p_t under exact components. The soft value is learned under this same law. QGPO's guidance scale rescales a learned energy gradient (Lu et al., 2023, Appendix G); it does not make this change to the conditional law. For fixed Q and λ, Proposition 1 preserves the terminal target across $\kappa \in (0, 1]$, although intermediate guidance changes. This uses the established reference-process construction cited in Appendix A.1; it does not imply that reducing κ always improves finite-sample estimation.

Adaptive temperature as an implementation choice. QGPO reports a fixed energy-model inverse temperature $\beta = 3$ (Lu et al., 2023, Appendix I.2). SVFM uses $\lambda_n = \kappa^2 s_{Q,n}/g$, with one scalar shared across states and flow times within update n (Appendix E). This rule adapts to continuation-critic dispersion; state-dependent or adaptive temperatures are also compatible with CEP and are not an exclusive capability of SVFM. Since λ_n also depends on κ, changing κ in this implementation need not preserve the terminal target. Our network tracks the changing target without taking λ as an input, so the fixed-objective guarantee does not establish convergence of adaptive training.

M Additional Analysis

M.1 Complementarity with Advanced Critic Learning

To test complementarity with advanced critic learning, we combine SVFM's policy improvement with FAC's flow-based critic regularization (Chae et al., 2026). We follow FAC's published evaluation protocol and domain-specific critic settings (Appendix F); FAC itself uses its released implementation and published hyperparameters unchanged.

Hyperparameter selection. With FAC's published settings fixed, we independently search the SVFM-specific parameters $\kappa \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and $g \in \{0.25, 0.5\}$ for SVFM and SVFM+FAC in each environment. Table 18 lists the selected configurations.

Table 18: Selected (κ, g) for the two Soft-Value Flow Matching columns of Table 20. The two searches are independent; both run under the protocol of Chae et al. (2026).

	SVFM		SVFM+FAC	
Environment	κ	g	κ	g
antmaze-large	0.2	0.25	0.4	0.25
antmaze-giant	0.4	0.5	0.4	0.5
humanoidmaze-medium	0.6	0.25	0.6	0.25
humanoidmaze-large	0.8	0.25	0.6	0.25
antsoccer	0.8	0.5	0.6	0.25
scene	0.6	0.25	0.6	0.25
puzzle-3x3	0.2	0.25	0.6	0.25
puzzle-4x4	0.6	0.25	0.4	0.25
cube-single	0.8	0.5	0.8	0.5
cube-double	0.4	0.25	0.4	0.25

Sensitivity to (κ, g). Table 19 reports the success rate of SVFM for every (κ, g) of the search range under the protocol of Chae et al. (2026).

Table 19: Success rate (%) of SVFM for each (κ, g) under the protocol of Chae et al. (2026), averaged over the 5 tasks of each domain, with 1 seed.

	$g = 0.25$					$g = 0.5$				
Environment $\setminus$ κ	0.2	0.4	0.6	0.8	1.0	0.2	0.4	0.6	0.8	1.0
antmaze-large	86.8	77.2	36.7	17.7	9.9	61.3	50.5	55.7	59.5	38.1
antmaze-giant	16.8	16.7	3.1	5.5	2.1	2.9	19.9	22.1	11.9	4.9
humanoidmaze-medium	18.9	63.1	83.5	81.1	74.0	0.0	18.5	58.3	70.9	79.1
humanoidmaze-large	0.0	2.3	10.4	25.7	14.8	0.0	0.0	1.7	2.9	10.9
antsoccer	27.5	53.6	56.9	44.9	38.0	0.8	32.0	50.7	57.2	53.7
scene	55.5	75.7	74.3	70.5	53.3	20.9	58.0	66.8	76.3	68.9
puzzle-3x3	92.4	85.7	86.3	12.8	6.1	71.6	80.1	78.7	81.2	20.7
puzzle-4x4	2.9	19.7	26.9	12.4	3.7	0.5	2.3	26.5	26.0	6.8
cube-single	78.8	88.4	92.3	91.9	55.9	9.7	80.1	91.7	94.8	81.3
cube-double	1.6	45.7	25.9	21.6	13.1	0.3	1.5	33.2	27.5	16.1

Table 20 shows that SVFM+FAC achieves an average score of 60.3, compared with 54.5 for FAC, an improvement of 5.8. With independently tuned hyperparameters, SVFM alone reaches 58.1 under the same protocol, exceeding FAC by 3.6, while SVFM+FAC scores a further 2.2 higher than SVFM alone. These results demonstrate that SVFM's actor remains effective when paired with FAC's critic learning rule, supporting its applicability beyond the critic configuration used in our main experiments.

Table 20: Performance of SVFM with improved critic learning (Chae et al., 2026). Scores are averaged over 8 seeds, with ± denoting the standard deviation. Bold indicates the highest mean score.

Environment (5 tasks)	FAC	SVFM	SVFM+FAC
antmaze-large	**80.1** ±10.5	77.5 ±13.8	75.0 ±18.6
antmaze-giant	**16.4** ±4.6	**16.4** ±8.9	15.8 ±9.5
humanoidmaze-medium	56.1 ±9.9	82.5 ±7.2	**88.8** ±3.7
humanoidmaze-large	4.3 ±2.3	21.6 ±4.5	**26.4** ±3.2
antsoccer	**63.8** ±2.1	59.3 ±2.6	57.6 ±1.9
scene	69.4 ±6.6	**76.4** ±1.2	76.2 ±1.6
puzzle-3x3	**99.6** ±0.2	86.3 ±6.7	89.4 ±1.7
puzzle-4x4	26.9 ±4.2	28.6 ±2.3	**37.1** ±2.3
cube-single	**98.4** ±0.4	94.4 ±0.8	96.0 ±1.0
cube-double	30.2 ±4.4	38.5 ±4.0	**40.7** ±6.5
Average Score	54.5 ±1.8	58.1 ±2.0	**60.3** ±2.3

M.2 Behavior Continuations versus Actor Continuations

SVFM draws the K continuations that define its soft-value target from the frozen behavior flow. We compare it with a variant that draws them from the current actor instead, keeping every other component and hyperparameter of Table 9 unchanged. Both variants are trained on the default task of each domain with 3 seeds (10001, 20002, 30003). Table 21 reports the success rate and the fraction of states at which the actor-supervision queries are valued by the critic above all 32 behavior candidates, as in Appendix I.2.

With continuations from the current actor, this fraction increases in every domain, from 1.6% to 5.6% on average, since the critic is now queried where the actor moves. At such states, the critic values the queried actions above every behavior-like action, so a higher fraction means that the actor is supervised more often at actions valued beyond the behavior support, which is the effect that the fixed behavior continuations are designed to avoid. The average success rate decreases from 56.3% to 51.3%. The largest drops occur on puzzle-3x3 ($65.3\% \rightarrow 36.0\%$), puzzle-4x4 ($21.3\% \rightarrow 10.0\%$) and humanoidmaze-large ($26.0\% \rightarrow 12.0\%$), while the two variants are within the seed variation on most other domains.

Table 21: Continuations from the current actor versus the behavior flow (SVFM), default task of each domain, 3 seeds. Success rate (%) is the mean $\pm$ standard deviation over seeds; queries above all candidates is the fraction (%) of states at which the actor-supervision queries are valued above all 32 behavior candidates.

	Continuations from the actor		Continuations from the behavior flow (SVFM)	
Environment	Success rate	Queries above all candidates	Success rate	Queries above all candidates
antmaze-large	88.7 ±0.9	3.6	**93.3** ±2.5	1.0
antmaze-giant	**5.3** ±4.1	7.5	0.0 ±0.0	1.6
humanoidmaze-medium	86.0 ±15.7	4.3	**94.0** ±1.6	1.3
humanoidmaze-large	12.0 ±8.5	3.2	**26.0** ±9.1	1.3
antsoccer	**36.7** ±1.9	3.8	33.3 ±5.7	1.4
scene	**97.3** ±2.5	9.8	90.7 ±9.3	1.8
puzzle-3x3	36.0 ±11.8	5.6	**65.3** ±14.7	1.9
puzzle-4x4	10.0 ±5.7	6.6	**21.3** ±5.7	1.8
cube-single	**98.0** ±2.8	5.5	95.3 ±3.4	2.0
cube-double	43.3 ±10.5	5.6	**44.0** ±18.5	1.8
Average	51.3 ±2.5	5.6	**56.3** ±2.9	1.6

M.3 Sensitivity to the Guidance Cap and the Minimum Flow Time

We vary two fixed settings of SVFM (Table 7) one at a time: we remove the guidance cap, and we set the minimum flow time to $t_{\min} = 0.05$ or 0.2 instead of 0.1. All other hyperparameters follow Table 9. Each variant is trained on the default task of each domain with 3 seeds. Table 22 reports the success rates.

Removing the guidance cap lowers the success rate on all three domains, from 67.6% to 62.0% on average. SVFM is insensitive to a smaller minimum flow time (67.3% with $t_{\min} = 0.05$). With $t_{\min} = 0.2$, the average is similar (66.7%), but the success rate decreases on puzzle-3x3 and increases on cube-double.

M.4 Number of Continuations

We vary the number of behavior continuations K per anchor, keeping all other hyperparameters of Table 9. Each variant is trained on the default task of each domain with 3 seeds. Table 23 reports the success rates.

With $K = 1$, SVFM fails on every domain. The relative temperature $\lambda = \kappa^2 s_Q / g$ uses the spread s_Q of critic values over the K continuations of an anchor, which is zero for a single continuation. The temperature then collapses to its numerical lower bound, so the guidance coefficient, which scales as $1/\lambda$, becomes very large and the guidance reaches the cap at every update. In addition, the soft-value target reduces to the critic value of one sampled continuation, so this maximal guidance follows a single noisy sample rather than an average over continuations. $K \geq 2$ is therefore required

Table 22: Success rate (%) of SVFM when one fixed setting is changed at a time: removing the guidance cap (with $t_{\min} = 0.1$), or changing $t_{\min}$ (with the guidance cap). Default task of each domain, mean ± standard deviation over 3 seeds.

	SVFM	Variants		
Guidance cap	✓	–	✓	✓
$t_{\min}$	0.1	0.1	0.05	0.2
antmaze-large	93.3 ±2.5	90.7 ±0.9	93.3 ±3.4	90.7 ±2.5
puzzle-3x3	65.3 ±14.7	60.0 ±4.9	68.7 ±3.8	50.7 ±7.4
cube-double	44.0 ±18.5	35.3 ±0.9	40.0 ±11.8	58.7 ±4.7
Average	67.6	62.0	67.3	66.7

for the relative temperature to be defined. With $K = 2$, the success rate is close to that of $K = 8$ on antmaze-large and scene, but decreases on the other three domains, and the average drops from 77.5% to 63.5%.

Table 23: Success rate (%) of SVFM with K behavior continuations per anchor, default task of each domain, mean ± standard deviation over 3 seeds. We use $K = 8$ by default.

Environment	$K = 1$	$K = 2$	$K = 8$
antmaze-large	0.0 ±0.0	91.3 ±0.9	93.3 ±2.5
humanoidmaze-medium	0.0 ±0.0	46.0 ±22.9	94.0 ±1.6
scene	0.0 ±0.0	98.7 ±1.9	90.7 ±9.3
puzzle-3x3	0.0 ±0.0	53.3 ±16.0	65.3 ±14.7
cube-double	0.0 ±0.0	28.0 ±15.0	44.0 ±18.5
Average	0.0	63.5	77.5

M.5 Wall-Clock Training Time

We measure the training and inference cost of each method on an NVIDIA RTX 3090 GPU, one method per GPU, with the hyperparameters of the *standard* setting and a batch size of 256. The time per gradient update includes sampling the batch and is the median over 4 trials of 250 updates after compilation. The inference latency is the time to generate one action for a single observation. Table 24 reports the results on cube-double and humanoidmaze-medium.

SVFM takes 5.0–5.2 ms per update, about 1.4 hours per 1M updates. This is about $2.1\times$ Q-Flow and $1.4\times$ QAM, and about 35% less than CEP. Table 25 breaks one update into its phases. Most of the additional cost over Q-Flow comes from the K behavior continuations: generating the soft-value targets takes 59% of an update on cube-double, against one actor rollout for Q-Flow's value target. The remaining phases take about the same time in both methods. Table 26 gives the same breakdown for the other methods: the largest phases are backpropagation through the flow for FBRAC (56%), the energy network and its labels over 32 support actions for CEP (67%), and the adjoint computation for QAM (59%). At inference, SVFM samples actions with the flow actor alone, without continuations or evaluations of the critic and the soft-value network, so its latency (0.6 ms) is close to that of Q-Flow, FBRAC and QAM, and about half that of CEP and Best-of-N ($N = 8$), which evaluate learned networks during sampling.

Table 24: Wall-clock cost on an RTX 3090 GPU: time per gradient update, hours per 1M updates and inference latency per action (single observation), on cube-double / humanoidmaze-medium. Best-of-N samples 8 actions from the behavior flow and selects the one with the highest critic value; it has no training phase of its own.

Method	ms / update	hours / 1M updates	Inference (ms)
Best-of-N	–	–	1.15 / 1.13
FBRAC (Park et al., 2025)	2.66 / 2.81	0.74 / 0.78	0.51 / 0.51
IFQL	1.77 / 1.96	0.49 / 0.55	0.65 / 0.83
FQL	1.80 / 1.99	0.50 / 0.55	0.21 / 0.23
CEP (Lu et al., 2023)	7.71 / 8.02	2.14 / 2.23	1.25 / 1.29
QAM (Li & Levine, 2026)	3.60 / 3.78	1.00 / 1.05	0.57 / 0.58
Q-Flow (Doo et al., 2026)	2.36 / 2.56	0.66 / 0.71	0.56 / 0.56
SVFM (ours)	5.00 / 5.23	1.39 / 1.45	0.62 / 0.61

Table 25: Time (ms) of each phase of one gradient update of Q-Flow and SVFM on cube-double / humanoidmaze-medium. Each phase is timed separately; because the compiler fuses computation across phases within an update, the sum of phases is slightly larger than the time per update in Table 24.

Phase	Q-Flow	SVFM (ours)
Value target (K behavior continuations / one actor rollout)	0.68 / 0.72	3.18 / 3.36
Value network regression	0.25 / 0.26	0.25 / 0.26
Critic target	0.67 / 0.73	0.64 / 0.71
Critic regression	0.25 / 0.25	0.25 / 0.25
Actor guidance target	0.19 / 0.19	0.20 / 0.20
Actor regression	0.18 / 0.19	0.18 / 0.19
Optimizer and target networks	0.53 / 0.56	0.65 / 0.68
Sum of phases	2.75 / 2.90	5.35 / 5.67

Table 26: Time (ms) of each phase of one gradient update of the other methods on cube-double / humanoidmaze-medium, and the share (%) of each phase on cube-double. Q-Flow and SVFM are in Table 25. Phases are timed separately, as in Table 25.

Phase	ms	%
FBRAC (Park et al., 2025)		
Critic target (10-step actor flow)	0.60 / 0.72	20
Critic regression	0.24 / 0.25	8
Behavior flow loss	0.18 / 0.19	6
Actor loss (BPTT through the 10-step flow)	1.71 / 1.85	56
Optimizer and target networks	0.31 / 0.33	10
Sum of phases	3.05 / 3.34	100
IFQL		
Value expectile regression	0.29 / 0.31	24
Critic regression	0.32 / 0.33	26
Behavior flow loss	0.18 / 0.19	15
Optimizer and target networks	0.41 / 0.44	34
Sum of phases	1.21 / 1.27	100
FQL		
Critic target (one-step policy)	0.15 / 0.16	8
Critic regression	0.24 / 0.25	13
Behavior flow loss	0.19 / 0.20	10
Distillation target (10-step behavior flow)	0.60 / 0.58	32
One-step actor loss	0.35 / 0.36	18
Optimizer and target networks	0.36 / 0.38	19
Sum of phases	1.91 / 1.93	100
CEP (Lu et al., 2023)		
Energy labels (target Q on 32 support actions)	1.68 / 1.77	24
Energy network regression (32 support actions)	3.05 / 3.11	43
Critic target (in-support backup, 32 actions)	1.68 / 1.78	24
Critic regression	0.24 / 0.25	3
Optimizer and target networks	0.43 / 0.45	6
Sum of phases	7.08 / 7.35	100
QAM (Li & Levine, 2026)		
Critic target (10-step flow)	0.69 / 0.65	17
Critic regression	0.25 / 0.25	6
Base flow loss	0.18 / 0.19	4
Adjoint states (10-step SDE, backward pass)	1.50 / 1.59	36
Adjoint-matching regression	0.95 / 1.04	23
Optimizer and target networks	0.55 / 0.58	13
Sum of phases	4.12 / 4.30	100

N LIBERO AND LIBERO-PLUS EXPERIMENTAL DETAILS

Benchmarks. LIBERO (Liu et al., 2023a) consists of four task suites, *Spatial*, *Object*, *Goal*, and *Long*, each with 10 language-conditioned manipulation tasks on a simulated Franka arm. We use the 5-shot setting: 5 human demonstrations per task, i.e., 200 demonstrations in total across the 40 tasks. We evaluate each policy on 50 episodes per task (2,000 episodes in total), with the standard LIBERO initial states and the standard per-suite episode lengths. LIBERO-Plus (Fei et al., 2025) evaluates the same 40 tasks under 10,030 perturbed configurations spanning seven perturbation types: camera viewpoint (1,599), robot initial state (1,550), language instruction (1,537), lighting (1,142), background texture (1,076), sensor noise (1,601), and object layout (1,525). Each configuration is evaluated once. All reported numbers are the mean and standard deviation over three evaluation seeds of the same checkpoint.

Why the 5-shot setting. With the full LIBERO training set (50 demonstrations per task), $\pi_{0.5}$ already reaches 96.9% average success (98.8%, 98.2%, 98.0%, and 92.4% on Spatial, Object, Goal, and Long), as reported with its open-source release (Black et al., 2025; Physical Intelligence, 2025). Its policy rollouts then contain almost no failures, which leaves little room for improvement and provides the critic with few unsuccessful outcomes to contrast against, so RL fine-tuning has little signal to learn from. Near this ceiling, differences between fine-tuning methods also fall within the evaluation noise. We therefore start from a BC policy trained on 5 demonstrations per task, which succeeds in 58% of the collected rollouts and leaves clear room for improvement, so that the effect of each method is measurable. Few-shot initializations are also the standard protocol of recent RL fine-tuning studies of $\pi_{0.5}$ on LIBERO (Wang et al., 2026).

BC policy. We use $\pi_{0.5}$ (Black et al., 2025) with an action chunk of $H = 10$ and 10 flow-integration steps. LIBERO uses 7 action dimensions (end-effector delta pose and gripper) of $\pi_{0.5}$'s 32 padded action dimensions. Starting from the pretrained $\pi_{0.5}$, we fine-tune the full model on the 200 demonstrations for 10K updates with learning rate 5×10^{-5} (1K warmup steps, constant afterwards) and the standard $\pi_{0.5}$ flow-matching loss. This BC policy is the initialization of every fine-tuning method, the policy used by the test-time methods, and the frozen base flow of SVFM. During evaluation, every policy executes the first 5 actions of each predicted chunk before replanning.

Policy rollouts. We collect 2,000 episodes with the BC policy (50 per task), executing each predicted chunk in full so that every stored chunk is the action that was actually executed. Half of the episodes start from a randomly reset scene, and the other half from intermediate states of the first half's rollouts; the benchmark evaluation initial states are never used for data collection. 1,162 episodes succeed and 838 time out. The reward is -1 per chunk and 0 on the chunk that completes the task, with discount $\gamma = 0.99$ per environment step (γ^H per chunk); a time-out is treated as truncation, not termination.

Critic. All critic-based methods use a chunk critic $Q(s, a)$ on frozen features of the BC policy: the mean-pooled prefix embedding of the vision-language model (2,048 dimensions) concatenated with the proprioceptive state. The critic is an ensemble of two MLPs with hidden sizes $[512, 512, 512]$ and layer normalization, uses the minimum over the target ensemble for bootstrapping, and updates its target network by EMA with rate 0.005. It is trained with TD learning on the demonstrations and all 2,000 rollouts, including failures, with learning rate 3×10^{-4} and batch size 64. Only the 7 real action dimensions are passed to the critic; the padded dimensions of sampled chunks are set to zero. The critic is first trained alone for 2K updates, with next actions sampled from the BC policy. SVFM then continues training it together with the actor for 5K updates, with the next action sampled from the current actor, i.e., 7K critic updates in total. To match this number of critic updates, BoN and QGF use the same critic after 5K additional critic-only updates (next action sampled from the BC policy), again 7K updates in total.

SVFM. The actor is initialized from the BC policy and fully fine-tuned (vision-language model and action expert, 3.6B parameters) for 5K updates with batch size 64, AdamW ($\beta = (0.9, 0.95)$), learning rate 2×10^{-5} (100 warmup updates, constant afterwards), and gradient clipping at 1. The actor data are the demonstrations and the 1,162 successful rollouts, sampled uniformly over transi-

tions; the frozen BC policy serves as the base flow. In each update, the critic and the soft-value head are updated first, and the actor then regresses the guided velocity target of Section 3 at flow times drawn from $\pi_{0.5}$'s own time distribution. We use $K = 8$ continuations for the soft value, $t_{\min} = 0.1$, the adaptive temperature with $(\kappa, g) = (0.6, 0.25)$, i.e., $c = \kappa^2/g = 1.44$, measured on each batch, and the bounded guidance with $\rho = 2$ applied to the 7 real action dimensions. We did not tune (κ, g) on LIBERO: we reuse the value selected on OGBench in the standard setting (Table 1; e.g., cube-single and antsoccer-arena), which is also the value used in the real-world experiments. Training takes about 7.4 hours on two NVIDIA B200 GPUs.

Baselines. *Filtered BC* continues BC on the same actor data (demonstrations and successful rollouts) with exactly the same initialization, batches, and schedule as SVFM, but with the plain flow-matching loss and no critic. It isolates the effect of the additional 5K updates on the successful rollouts from that of the critic. *BoN* samples 8 chunks from the BC policy at every replanning step and executes the one with the highest critic value (minimum over the ensemble). *QGF* (Zhou et al., 2026) uses the critic gradient to guide the sampling process of the BC policy at test time, with the same (κ, g), $t_{\min}$, and ρ; the temperature is fixed to the average adaptive temperature measured during critic training. *Q-VGM* (Wang et al., 2026) fine-tunes the actor with its critic-guided velocity-matching objective using the paper's hyperparameters: $M = 5$ guided late denoising steps, $J = 3$ normalized ascent steps of size 0.05 with keep-best selection, and gradient clipping at 3.5. Following the paper, it uses its own offline critic, trained first and then frozen: an IQL (Kostrikov et al., 2022) critic with 10 heads of widths $[512, 512, 256]$, the action re-injected at every hidden layer, expectile 0.8, and 6K updates on the same data and features. Q-VGM uses the same initialization, actor data, batch size, and number of updates as SVFM. Since the paper does not specify an offline actor learning rate, we use that of SVFM.

Reproducing Q-VGM. The Q-VGM paper reports that offline Q-VGM improves a few-shot $\pi_{0.5}$ policy from 84.6% to 90.7% on LIBERO, while Filtered BC reaches 85.1%. In our setting, Q-VGM improves BC by about 3 points but stays below Filtered BC. To check whether this is due to the critic, we ran Q-VGM with three critics (Table 27): the frozen 7K TD critic shared with BoN and QGF, the paper's IQL critic, and a critic trained jointly with the actor exactly as in SVFM. All three give 78–80% on LIBERO and 66–67% on LIBERO-Plus. We also verified our implementation numerically against the paper's equations, including the time and sign conventions of the look-forward and the velocity target.

Table 27: Success rates (%) of Q-VGM with different critics on LIBERO and LIBERO-Plus. Mean $\pm$ standard deviation over 3 evaluation seeds, except where noted.

Critic	Critic updates	LIBERO	LIBERO-Plus
TD critic, frozen (shared with BoN and QGF)	7K	77.5 (2 seeds)	–
IQL critic, frozen (paper's setting)	6K	79.7 ± 0.8	67.4 ± 0.3
TD critic, trained jointly with the actor (as in SVFM)	2K + 5K	78.4 ± 0.6	66.5 ± 0.6
BC	–	76.8 ± 0.4	65.9 ± 0.5
Filtered BC	–	81.7 ± 0.4	69.3 ± 0.3
SVFM	2K + 5K	86.4 ± 0.8	72.9 ± 0.6

We attribute the gap to the paper mainly to differences in the setting. Most importantly, Filtered BC is a much stronger baseline in our setting: it improves BC by 4.9 points here, compared with 0.5 points in the Q-VGM paper. Our actor data contain the 200 demonstrations and 1,162 successful rollouts, whereas the paper's offline buffer contains only 150 rollouts per suite collected with exploration noise and no demonstrations. Q-VGM has no imitation term: it uses only the states of the data and moves the actor along the critic's action gradient, so it cannot directly exploit the successful actions that drive the improvement of Filtered BC and SVFM. Other differences may also contribute: our chunk length is $H = 10$ rather than 5, so the fixed ascent step of 0.05 is spread over twice as many action dimensions; we fine-tune the full model rather than only the action expert; our critic uses the mean-pooled prefix features instead of the paper's learned RL token; our reward is -1 per chunk rather than sparse; and our rollouts are collected without exploration noise, which gives the critic less action diversity from which to learn action gradients. In all three variants, the keep-best gate

accepted every ascent step, so the target reduced to a fixed-length step along the normalized critic gradient.