FLOW POLICY OPTIMIZATION WITH STEERABLE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow policies offer an expressive and scalable foundation for continuous control, but online reinforcement learning becomes subtle when generated actions are refined before execution. An action with low current value may still be a valuable sample if a small edit can move it into a high-value region, so the generator should account for what its actions can become after correction. We introduce *Steerable Flow Editing* (SFEdit), a proximal framework that derives distribution learning and action refinement from a single policy-improvement objective. A Wasserstein-smoothed KL regularizer yields an exact *tilt–transport* factorization: the flow favors actions by the value attainable after correction, net of the movement cost, and a proximal map performs the correction. The scaled proximal displacement provides the actor learning signal, and the same map determines the actions used for interaction and critic backups. We realize the resulting tilt through adjoint matching and introduce iQAM, an integrating-factor discretization that reduces the degradation of few-step learning without additional vector–Jacobian products. On 50 OGBench tasks across ten domains, SFEdit achieves 87.6% final aggregate success, compared with 80.7% for the strongest baseline under the shared evaluation protocol. On two fine-grained real-robot manipulation tasks, it nearly triples the success of a frozen $\pi_{0.5}$ generalist within 45 minutes of online training.

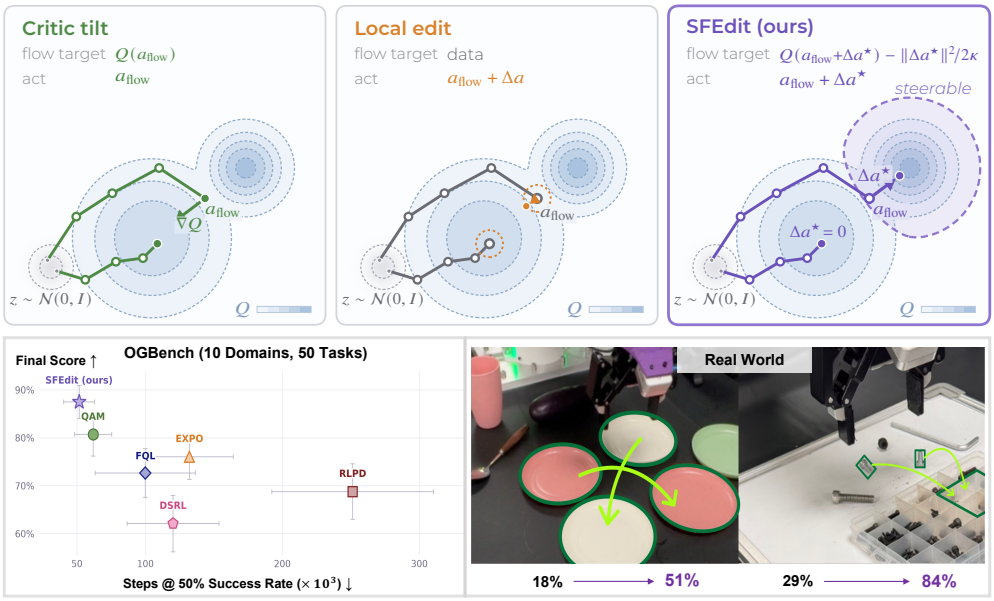

Figure 1: **SFEdit edits an action only when the value it reaches pays for the move.** Top: one critic Q with a broad mode (rings) and a narrower, higher mode (target). The critic tilt pulls the flow sample a into the broad mode, and a local edit corrects a within a fixed radius; SFEdit trains the flow on the proximal displacement and executes the edited action, which reaches the high mode from the steerable region (dashed). Bottom: final OGBench success against environment steps needed to first reach 50% success, and real-robot success of the frozen $\pi_{0.5}$ teacher (left) and of SFEdit (right).

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.

1 INTRODUCTION

Flow policies generate actions by integrating a learned velocity field from noise, which lets them model multimodal behavior and scale to generalist robot control (Lipman et al., 2023; Chi et al., 2023; Black et al., 2025b). However, imitation-based learning bounds a policy by the quality and coverage of its data. Online reinforcement learning (RL) can improve the policy beyond its data, provided that a critic's evaluation of individual actions can be turned into an update of the process that generated them.

Two mechanisms do this (Figure 1). *Tilting* trains the flow so that its samples favor actions the critic values, for example by regressing the velocity field onto critic-derived targets with adjoint matching (Domingo-Enrich et al., 2025), as in Q-learning with adjoint matching (QAM; Li & Levine, 2026). *Editing* corrects each sample before execution, with a learned editor or a few steps of critic ascent (Dong et al., 2026a; Sobol Mark et al., 2024). Tilting reallocates probability among actions the reference already produces, whereas editing moves individual actions, so the two are naturally combined (Li & Levine, 2026). Suppose the critic has a broad region of moderate value and a narrow peak of high value. An action just outside the peak has low value, so the tilt suppresses it, yet a short edit would carry it onto the peak. Once samples are edited, an action is worth generating for the value it reaches after the edit, not for the value it has, and tilting the flow by the critic trains it as if its samples were final. We therefore ask: *what should a flow learn when its actions will be edited?*

We answer with a single objective. Policy improvement is posed jointly over the law the flow generates and the law that is executed, with a KL penalty toward a reference and a quadratic cost on the movement between the two; eliminating the generated law leaves the Moreau–Yosida regularization of the KL divergence in Wasserstein space (Jordan et al., 1998; Baptista et al., 2026). For a fixed critic, the optimizer factorizes exactly (Theorem 1). The flow should learn the reference tilted by the *upper Moreau envelope* of the critic, which scores an action by the best value reachable from it net of the movement cost, and the executed policy is the image of this tilt under the associated *proximal map*. In the one-dimensional example of Figure 2, this tilt more than doubles the mass near the narrow peak, and after the same edit the expected executed value rises from 0.62 under the critic tilt to 0.76. Tilting by the unedited critic leaves a gap in this objective that no editor closes, and the gap vanishes only when every action gains equally from editing (Proposition 2).

We develop *Steerable Flow Editing* (SFEdit) around this factorization. Its practicality rests on one identity: the proximal displacement, divided by the movement scale, is the gradient of the envelope (Lemma 3). Editing an action and computing its learning signal are therefore the same computation, and one proximal operator supplies the actor's learning signal, the executed action, and the action at which the critic bootstraps. We learn the tilt with adjoint matching, which requires neither policy likelihoods nor backpropagation through sampling, and add iQAM, an integrating-factor discretization of the adjoint recursion that reduces the degradation of few-step adjoint learning without additional vector–Jacobian products (VJPs). Our contributions are:

- a variational account of flow editing, the exact tilt–transport solution of proximal policy improvement and the gap of tilting by the unedited critic, which together say what a flow should learn when its actions are edited;
- SFEdit, an online RL algorithm in which one proximal operator provides the learning signal, the executed action, and the bootstrap action, together with iQAM for few-step adjoint learning;
- an evaluation on 50 OGBench tasks against nine baselines with matched ablations of the target, the transport, and the discretization, and online adaptation of a frozen $\pi_{0.5}$ policy on two fine-grained real-robot tasks.

2 RELATED WORK

RL for generative policies. Diffusion and flow policies capture multimodal behavior and scale from visuomotor imitation to generalist robot control (Chi et al., 2023; Black et al., 2025b;a). RL improves them through value-guided training and distillation (Wang et al., 2023; Park et al., 2025b), policy gradients through the generation process (Ren et al., 2025; Zhang et al., 2025; McAllister et al., 2026), regression onto critic-derived targets (Psenka et al., 2024; Li & Levine, 2026), or by

steering a fixed prior through its latent noise or by re-ranking its samples (Wagenmaker et al., 2025; Im et al., 2026; Nakamoto et al., 2024). Action chunking (Li et al., 2025), few-step generation (Frans et al., 2025; Geng et al., 2025; Zhan et al., 2026), and compact interfaces to large models (Xu et al., 2026a) make such adaptation efficient. SFEdit belongs to the regression family but changes the target: the flow is tilted by the value its samples reach after correction.

Tilting and editing samples. Stochastic optimal control turns reward-tilted sampling into regression problems for diffusion and flow models (Domingo-Enrich et al., 2024; 2025; Havens et al., 2025; Domingo-Enrich & Han, 2026; Domingo-Enrich et al., 2026); related methods learn tilts without reward gradients (Potaptchik et al., 2026) or bound online updates by path-space trust regions (Dong et al., 2026b). A complementary line improves actions after they are generated, through learned bounded editors (Dong et al., 2026a; Li & Levine, 2026), critic ascent on sampled or replayed actions followed by distillation into the policy (Yang et al., 2023; Sobol Mark et al., 2024), or value-gradient guidance of a fixed generator (Xu et al., 2026b; Zhou et al., 2026). These methods differ from SFEdit in what the generator learns: it imitates data or improved actions, follows the unedited critic tilt, or stays fixed. The idea of training a model for the value it attains after an inference-time procedure also underlies inference-aware alignment of language models under best-of-N sampling (Balashankar et al., 2025); for a cost-penalized continuous edit, it takes the closed form of an envelope tilt.

Wasserstein geometry in policy optimization. Relative entropy controls how probability is reallocated in policy improvement (Peters et al., 2010; Abdolmaleki et al., 2018), whereas Wasserstein penalties charge for moving actions (Pfau et al., 2025). Their infimal convolution is the Moreau–Yosida regularization of the KL divergence in Wasserstein space, a proximal optimal-transport divergence (Moreau, 1965; Ambrosio et al., 2008; Baptista et al., 2026), and Theorem 1 is closely related to its dual representation. Variants of QAM already refine the samples of a KL-regularized tilt under a Wasserstein or ℓ_∞ bound, but take the intermediate law to be the ordinary critic tilt (Li & Levine, 2026). We solve the joint problem exactly, identify its two factors with a flow learner and an action editor, and quantify the gap left by the ordinary tilt.

3 Preliminaries

Online reinforcement learning. We consider a discounted MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$ with continuous actions $a \in \mathcal{A} \subset \mathbb{R}^d$, possibly chunks of primitive commands (Zhao et al., 2023), and seek a policy that maximizes the expected discounted return. An off-policy actor–critic learns an action-value function Q_ϕ from replay transitions $(s, a, r, s', m) \sim \mathcal{D}$ with the detached temporal-difference target

$$\mathcal{L}_{\mathrm{TD}}(\phi) = \mathbb{E}_{\mathcal{D},\, a' \sim \pi(\cdot|s')} \big[Q_\phi(s,a) - r - m\gamma \bar{Q}(s', a') \big]^2, \tag{1}$$

where m is the bootstrap mask and $\bar{Q}$ a target critic; Appendix G covers action chunks.

Conditional flow policies. A state-conditioned velocity field transports $x_0 \sim \mathcal{N}(0, I)$ to an action along $\dot{x}_t = v_\theta(x_t, t \mid s)$, $t \in [0,1]$, with endpoint law $\pi_\theta(\cdot \mid s)$. For the linear interpolation $x_t = (1-t)x_0 + ta$, flow matching minimizes

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D},\, x_0,\, t} \big\| v_\theta(x_t, t \mid s) - (a - x_0) \big\|^2. \tag{2}$$

A reference field v_{ref} with endpoint law π_{ref} is fixed within each improvement step. Action normalization and clipping are treated in Appendix D.3.

Adjoint matching for tilted flows. For a differentiable bounded utility f, the problem $\max_\mu \{\mathbb{E}_\mu f - \mathrm{KL}(\mu \| \pi_{\mathrm{ref}})\}$ has solution $d\mu \propto e^f d\pi_{\mathrm{ref}}$. Adjoint matching learns this tilt by regression on detached trajectories (Domingo-Enrich et al., 2025). For the linear interpolation, the memoryless stochastic representation uses

$$b_{\mathrm{ref}}(x,t) = 2v_{\mathrm{ref}}(x, t \mid s) - \frac{x}{t}, \qquad \dot{\lambda}_t = -\nabla_x b_{\mathrm{ref}}(x_t, t)^\top \lambda_t, \qquad \lambda_1 = -\nabla f(x_1), \tag{3}$$

with noise $\sigma_t = \sqrt{2(1-t)/t}$ for $0 < t < 1$. Training paths follow the current controlled field, backward Jacobians use the reference, and the regression loss is

$$\mathcal{L}_{\mathrm{AM}}(\theta) = \mathbb{E} \int_0^1 \left\| \frac{2\big(v_\theta(x_t, t \mid s) - v_{\mathrm{ref}}(x_t, t \mid s)\big)}{\sigma_t} + \sigma_t \lambda_t \right\|^2 dt. \tag{4}$$

Paths, adjoints, and reference values are detached. QAM uses $f = \beta Q$ with $\beta > 0$ (Li & Levine, 2026); Appendix D gives the continuous-time interpretation and endpoint conventions.

4 Steerable Flow Editing

SFEdit trains a flow to anticipate the correction applied to its actions. We first derive the target distribution, then show that its learning signal also determines the executed and bootstrapped actions.

4.1 Proximal policy improvement

Fix a state, which we suppress from the notation, together with a reference law π_{ref} and a continuous critic Q on a nonempty compact convex action set $\mathcal{A}$. We jointly optimize the intermediate law μ learned by the flow and the executed action law π:

$$\max_{\mu,\pi} \left\{ \beta \mathbb{E}_\pi Q - \mathrm{KL}(\mu \| \pi_{\mathrm{ref}}) - \frac{W_2^2(\mu,\pi)}{2\kappa} \right\}, \qquad \beta, \kappa > 0. \tag{5}$$

The KL term controls how probability is reallocated, and the Wasserstein term charges for moving actions. Eliminating μ gives a Wasserstein-smoothed KL regularizer that permits finite-cost movement beyond the reference support (Baptista et al., 2026); its definition and properties appear in Appendix A.1. For an anchor a, that is, an action proposed by the flow, define

$$U_\kappa(a) = \max_{b \in \mathcal{A}} \left\{ \beta Q(b) - \frac{\|b-a\|^2}{2\kappa} \right\}, \tag{6}$$

and let $T_\kappa(a)$ be a maximizing action. The upper Moreau envelope U_κ scores the best penalized correction, and the proximal map T_κ returns its endpoint (Moreau, 1965; Rockafellar & Wets, 1998).

Theorem 1 (Tilt–transport factorization). *For any measurable selection T_κ of maximizers in (6), an optimizer of (5) is*

$$\underbrace{d\mu^+(a) = Z^{-1} e^{U_\kappa(a)} d\pi_{\mathrm{ref}}(a)}_{\textit{envelope tilt}}, \qquad \underbrace{\pi^+ = (T_\kappa)_{\#}\mu^+}_{\textit{proximal transport}}, \quad Z = \mathbb{E}_{\pi_{\mathrm{ref}}} e^{U_\kappa}. \tag{7}$$

The optimal value is $\log Z$, and the intermediate law μ^+ is unique.

Here $T_{\#}\mu$ is the law of $T(a)$ for $a \sim \mu$. The proof optimizes transported endpoints pointwise, leaving a Gibbs variational problem over anchors (Appendix A). The envelope tilt allocates probability according to what the subsequent correction can achieve. The movement scale interpolates between tilting and editing: as $\kappa \to 0$, $U_\kappa \to \beta Q$ uniformly and T_κ tends to the identity, which recovers the ordinary critic tilt; as $\kappa \to \infty$, $\mu^+ \to \pi_{\mathrm{ref}}$ and every limit point of $T_\kappa(a)$ maximizes Q, so reference samples are moved greedily to critic maxima (Appendix B.3).

Proposition 2 (Gap of the unedited tilt). *Let $d\mu_Q \propto e^{\beta Q} d\pi_{\mathrm{ref}}$ and $\Delta = U_\kappa - \beta Q \geq 0$. The pair $(\mu_Q, (T_\kappa)_{\#}\mu_Q)$ falls short of the optimum in (5) by*

$$\mathrm{KL}(\mu_Q \| \mu^+) = \log \mathbb{E}_{\mu_Q} e^{\Delta} - \mathbb{E}_{\mu_Q} \Delta \geq 0, \tag{8}$$

with equality exactly when Δ is constant μ_Q-almost surely.

Here $\Delta(a)$ is the correction gain of an anchor, about $\kappa\beta^2 \|\nabla Q(a)\|^2/2$ for short edits, so the envelope tilt favors anchors on steep ascent directions. Even an exact editor leaves a gap when anchors differ in this gain (Figure 2), and for the anchor law μ_Q no other editor does better (Appendix C.4). The statement concerns the target law, not the flow learner.

4.2 One operator for learning, acting, and bootstrapping

For derivatives, extend (6) to anchors in $\mathbb{R}^d$. The tilt and transport then share a simple identity.

Lemma 3 (Envelope gradient). *If the maximizing endpoint in (6) is unique at a, then U_κ is differentiable there and*

$$\nabla U_\kappa(a) = \frac{T_\kappa(a) - a}{\kappa}. \tag{9}$$

Appendix B.1 derives it from Danskin's theorem (Danskin, 1966). For actor learning, substituting $f = U_\kappa$ in (3) gives the terminal signal $\lambda_1 = -(T_\kappa(x_1) - x_1)/\kappa$. For interaction, the policy executes $T_\kappa(a)$ for $a \sim \pi_\theta$ and stores it in replay. For bootstrapping, the same map is applied to the next-state sample in (1), so the critic evaluates the executed policy.

Proposition 4 (Fixed-critic improvement). *The optimizer in Theorem 1 satisfies*

$$\mathbb{E}_{\pi^+} Q - \mathbb{E}_{\pi_{\text{ref}}} Q \geq \frac{\mathrm{KL}(\mu^+ \| \pi_{\text{ref}})}{\beta} + \frac{\mathbb{E}_{a\sim\mu^+} \|T_\kappa(a) - a\|^2}{2\kappa\beta}. \tag{10}$$

If Q is continuously differentiable near $\mathcal{A}$ and $\|\nabla Q\| \leq G$ on $\mathcal{A}$, then $\|T_\kappa(a) - a\| \leq \kappa\beta G$ for every $a \in \mathcal{A}$.

Both terms on the right are nonnegative: the gain in critic value is at least the KL cost of reallocating probability plus the cost of moving actions. The bound concerns an exact update under a fixed critic; approximate actor fitting, numerical sampling, and finite proximal solves are analyzed in Appendix C, and no guarantee of monotonic improvement in online return follows.

4.3 Integrating-factor adjoint discretization

For a sampled path, write $J(t) = \nabla_x v_{\text{ref}}(x_t, t \mid s)$ and $\tau(t) = t + \varepsilon > 0$ for the numerical denominator. The adjoint obeys $\dot\lambda = -2J^\top\lambda + \lambda/\tau$. Setting $\lambda = \tau z$ removes the scalar term, in the spirit of the exponential integrators used for diffusion sampling (Zhang & Chen, 2023; Lu et al., 2022). An explicit Euler step in z, from t_{i+1} to t_i, gives

$$\lambda_i = d_i\lambda_{i+1} + d_i\, 2h_i J_{i+1}^\top \lambda_{i+1}, \qquad d_i = \frac{\tau_i}{\tau_{i+1}}, \quad h_i = t_{i+1} - t_i. \tag{11}$$

The violet factor is the only change from direct Euler in λ: it carries the known scalar attenuation into the learned-Jacobian increment, using the same VJP, so its weight is fixed by the dynamics rather than tuned. Both schemes are exact when $J = 0$; Proposition 5 in Appendix E gives the resulting norm bound and the scope of its quadrature advantage. SFEdit uses iQAM by default.

4.4 Practical algorithm

For box-bounded action coordinates, we approximate T_κ by the relaxed projected fixed-point iteration

$$y_0 = a, \qquad y_{j+1} = \mathrm{clip}\left(y_j + \rho\,(a + \kappa\beta\nabla Q(y_j) - y_j),\, a_{\min}, a_{\max}\right), \qquad \widehat{T}_\kappa(a) = y_M, \tag{12}$$

with the anchor a fixed and $\rho = 1/2$. Its fixed points are exactly the stationary points of (6), and it converges to the unique maximizer with factor $1 - \rho(1 - \kappa\beta L)$ if ∇Q is L-Lipschitz and $\kappa\beta L < 1$; outside this regime, even a converged solve is only guaranteed to be a stationary point (Appendix C). One iteration with inactive clipping returns the raw critic gradient scaled by ρ, so isolating the envelope target requires $M \geq 2$ (Appendix C.2); we use $M = 4$ throughout.

Algorithm 1 applies $\widehat{T}_\kappa$ at all three sites, with the reference and target critics held fixed while targets are built; paths, proximal iterates, and adjoint targets are detached. The actor loss adds replay flow matching to the adjoint term, which keeps the flow close to the data, so the learned law trades off μ^+ against the replay distribution rather than matching μ^+ exactly. Each proximal call costs M critic-gradient evaluations, and iQAM costs none; Appendix G lists all settings.

5 Experiments

We first examine the exact solution of (5) in one dimension and then ask whether SFEdit improves flow policies through online interaction, which of its components account for the gains, and whether the gains carry over to fine-grained real-robot manipulation. Throughout, *SFEdit* uses iQAM;

Algorithm 1 SFEdit. Violet marks the envelope targets, transport, and iQAM factor.

Input: Actor v_θ, reference v_{ref}, target critics, replay $\mathcal{D}$; β, κ; steps K, M

1: Sample $\tilde{a} \sim \pi_\theta(\cdot \mid s)$; execute $a = \widehat{T}_\kappa(s, \tilde{a})$; store (s, a, r, s', m)
2: Sample a replay batch and $\tilde{a}' \sim \pi_\theta(\cdot \mid s')$; set $a' = \widehat{T}_\kappa(s', \tilde{a}')$; update critics by (1)
3: Sample a detached K-step training path under the current controlled field
4: Set $\lambda_K = -\big(\widehat{T}_\kappa(s, x_K) - x_K\big)/\kappa$
5: **for** $i = K-1, \ldots, 0$ **do**
6: $\lambda_i = d_i \lambda_{i+1} + d_i \, 2h_i \, \mathrm{VJP}(v_{\mathrm{ref}}, x_{i+1}, \lambda_{i+1})$
7: **end for**
8: Update the actor on $\mathcal{L}_{\mathrm{FM}} + c_{\mathrm{AM}}\mathcal{L}_{\mathrm{AM}}$; update target critics and reference

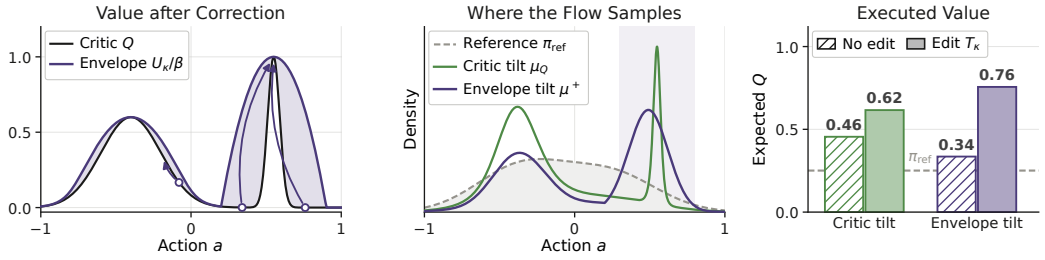

Figure 2: **The flow should favor actions that the edit can improve** (exact solution of (5) in one dimension, $\beta = 3$, $\kappa = 0.02$). Left: the envelope U_κ/β exceeds the critic by the correction gain Δ/β (shaded), and arrows show three proximal edits. Middle: within the shaded band around the narrow peak, the mass rises from 0.23 under the critic tilt μ_Q to 0.49 under the envelope tilt μ^+. Right: expected value of the executed action; the gap of Proposition 2 is 0.27 nats.

SFEdit(QAM) keeps the original adjoint discretization with the same envelope target and transport; and *iQAM* changes only the discretization of QAM.

A one-dimensional example. Figure 2 solves (5) exactly for a critic with a broad mode and a narrow, higher peak. The envelope tilt moves mass toward actions from which a short edit reaches the peak; with the edit, the expected value rises from 0.62 under the critic tilt to 0.76, and without it, the envelope tilt falls to 0.34, below the critic tilt alone. To reach 0.76 with the same edit, the critic tilt needs twice the inverse temperature and more than twice the regularization cost (Appendix C.4), so the envelope tilt is not merely a more aggressive critic tilt.

5.1 Experimental setup

Simulation. We evaluate 50 tasks, five in each of ten OGBench domains spanning maze navigation, sequential manipulation, and combinatorial rearrangement (Park et al., 2025a): antmaze (large, giant), humanoidmaze (medium, large), cube (double, triple, quadruple), puzzle (3×3, 4×4), and scene. So that online interaction carries the improvement, every method is initialized from the same per-task checkpoint of offline QAM with at most 10% success and then trains for 500k environment steps with one update per step. We run five seeds per task and report aggregate success over the 50 tasks. All tilt-based methods, including QAM, share the inverse temperature $\beta = 3$, which QAM's own protocol also uses in five domains including antmaze-giant, and SFEdit uses $\kappa = 0.02$ in nine domains and 0.2 in antmaze-giant; checkpoint selection, evaluation, and aggregation are specified in Appendix I.

Real world. We use a Franka Research 3 arm with a wrist-mounted and an external camera and design two tasks that demand precise, multi-stage manipulation (Figure 5). *Screw Sorting* requires repeatedly picking screws of a target size among others and placing them into empty slots of a screw box, which demands accurate alignment, well-timed grasps, and precise placement. *Plate Stacking* requires stacking plates by color: the shallow plates must be grasped at accurate approach angles, and the next plate must be chosen among distractors, including cups, spoons, and plates of colors absent from the demonstrations. Object positions are randomized, and an episode succeeds only if the task

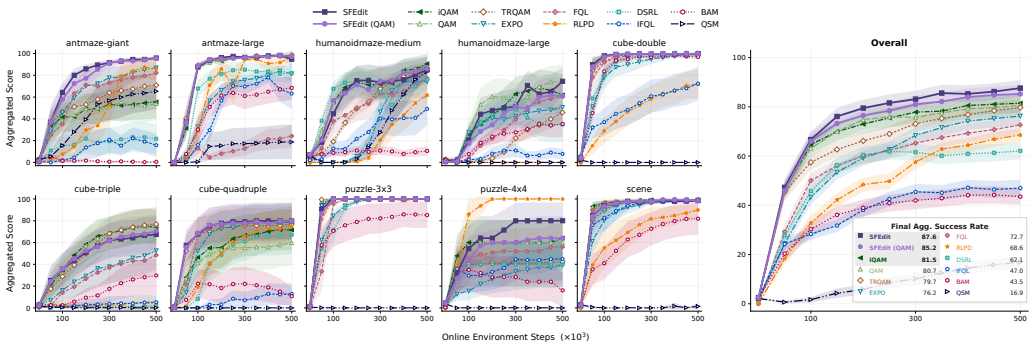

Figure 3: **Online learning on ten OGBench domains (five tasks each, five seeds).** The rightmost panel averages the ten domains, and its legend lists the final aggregate success of each method. SFEdit uses iQAM, while SFEdit(QAM) retains the original adjoint update. Shaded bands are 95% intervals over the 25 seeds of a domain and, for the aggregate, the propagated domain standard errors.

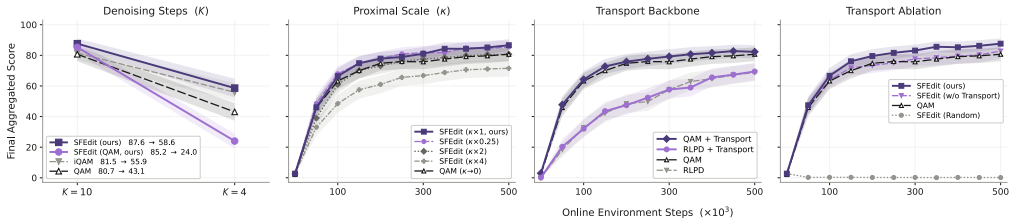

Figure 4: **Ablations (ten domains, five seeds).** From left: final aggregate success with ten or four flow steps; movement scale κ, with QAM for reference; transport added to QAM and RLPD; and transport removed or randomized in direction, both built on SFEdit(QAM). Bands are ± 1 standard error across tasks.

is completed within 200 control steps for Screw Sorting and 250 for Plate Stacking. The teacher is $\pi_{0.5}$ (Black et al., 2025a) fine-tuned on 100 teleoperated demonstrations (60 for Screw Sorting and 40 for Plate Stacking); its success has plateaued under this imitation recipe, and it stays frozen during online learning. Each method receives three runs per task, each with a 45-minute online budget and no human interventions, and each final policy is evaluated on 30 episodes (Appendix K).

Student policy. For SFEdit and QAM, we distill from the teacher a compact flow with about $1/50$ of the parameters of its action expert, which keeps online updates cheap. The student takes the teacher's frozen representation and proprioception as input, generates an action chunk with four Euler steps, and predicts the action rather than a residual; only the student and the critic are trained online.

Baselines. In simulation, we compare with *RLPD*, a Gaussian actor–critic using prior data (Ball et al., 2023); *FQL*, which distills a flow into a one-step actor (Park et al., 2025b); *IFQL*, the flow counterpart of implicit Q-learning policy extraction (Hansen-Estruch et al., 2023; Li & Levine, 2026); *DSRL*, which optimizes the sampler's latent noise (Wagenmaker et al., 2025); *EXPO*, which learns edits to an expressive base policy (Dong et al., 2026a); *QSM*, which supervises denoising with critic gradients (Psenka et al., 2024); and *QAM*, its basic-adjoint ablation *BAM*, and its trust-region variant *TRQAM* (Li & Levine, 2026; Dong et al., 2026b). On the real robot, DSRL steers the frozen teacher through its initial noise; *RLT* trains a compact actor–critic on the teacher's representation and reference action chunks (Xu et al., 2026a), as an implicit case of EXPO-style method; and QAM uses the same student as SFEdit with the unedited critic tilt, the original adjoint discretization, and no transport. Within each setting, all methods share data, evaluation, and online budget, while policy interfaces follow each method (Appendix H).

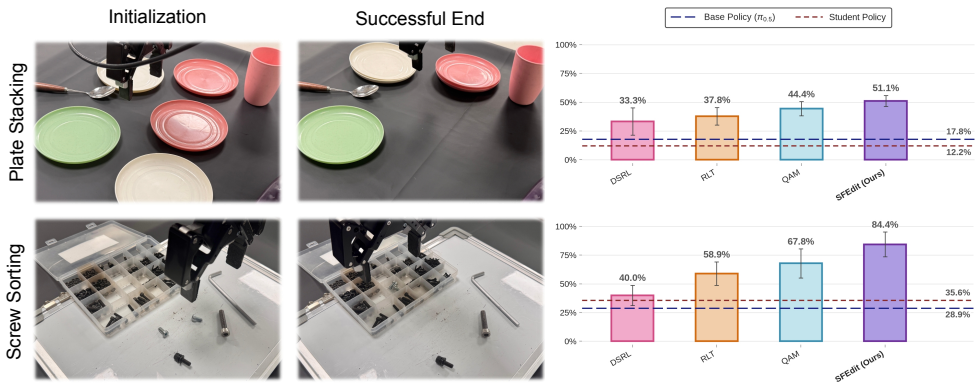

Figure 5: **Real-robot tasks and success after online adaptation.** Left: initial and completed scenes. Right: success after the 45-minute online training budget, averaged over three runs with 30 evaluation episodes each; error bars indicate variation across runs. Dashed lines mark the frozen $\pi_{0.5}$ teacher (Base Policy) and the distilled student before RL.

5.2 Online learning in simulation

Results. SFEdit reaches the highest final aggregate success in Figure 3 with 87.6%, and reaches 50% success earliest on average, after about 52k steps (Figure 1). Its gains over QAM concentrate in antmaze-giant, where the shared temperature is QAM's own and SFEdit adds 42 points, and in cube-quadruple and puzzle-4×4, where it adds 20 points each, while RLPD alone solves puzzle-4×4 and QAM and TRQAM finish higher on cube-triple. Most of the advantage comes from the target and the transport: SFEdit(QAM), which keeps the adjoint update of QAM, gains 4.5 points over QAM and exceeds every baseline, whereas iQAM alone adds 0.8 points.

Behavior analysis and ablations. In the last two panels of Figure 4, QAM+transport edits the samples of the ordinary critic tilt with the same solver, which approximates the best editor for that anchor law (Appendix C.4); replacing its critic target by the envelope target, which yields SFEdit(QAM), raises the aggregate from 82.3 to 85.2 (Appendix J). The two factors matter most where corrections are large: in antmaze-giant, whose movement scale is ten times larger, the envelope target without transport falls to 38.2, below QAM's 53.9, whereas the combination reaches 95.4, the pattern of Figure 2 at benchmark scale. Elsewhere, removing the transport shifts each domain by a few points, within the spread between QAM and iQAM, and raises the nine-domain average by 3.7 points. A random direction of the same magnitude destroys learning, so the transport acts through its direction rather than as exploration noise. Movement also needs a price: a quarter of the default κ performs about as well, whereas doubling it falls to the level of QAM and quadrupling it degrades learning further, as large edits leave the region where the critic was trained. With four instead of ten flow steps, the original adjoint recursion degrades sharply, most of all with the envelope target, and iQAM substantially reduces this degradation at the same number of VJPs.

5.3 Real-robot adaptation

Results. The real-robot study asks whether online learning can lift a generalist policy beyond its imitation plateau. On Screw Sorting, SFEdit reaches 84.4% success after 45 minutes of online training, against 28.9% for the frozen teacher and 35.6% for the distilled student, so the gain is made online rather than by distillation (Figure 5). Under the same budget, QAM reaches 67.8%, RLT 58.9%, and DSRL 40.0%. On Plate Stacking, SFEdit raises success from 17.8% for the teacher to 51.1%, ahead of QAM at 44.4%, RLT at 37.8%, and DSRL at 33.3%. With three runs per method, the error bars of SFEdit clear those of RLT and DSRL on Screw Sorting but overlap those of QAM on both tasks. QAM shares the student and critic but none of the envelope target, the transport, or iQAM, so this comparison measures the full update, including iQAM at the student's four flow steps. In our trials, the gain on Plate Stacking comes mainly from grasping the shallow plates at accurate angles, a local correction that the transport can supply and help optimize.

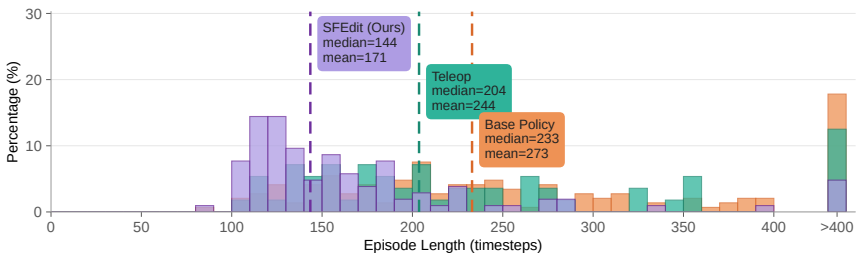

Figure 6: **Episode lengths of successful Screw Sorting rollouts with the step limit lifted.** Medians (dashed): SFEdit 144 steps, teleoperation (Teleop) 204, frozen teacher (Base Policy) 233. Each histogram is normalized within its method, and the last bin pools episodes longer than 400 steps.

Case study. Because success is counted within a fixed step budget, part of these gains reflects faster completion; to separate and further compare the execution efficiency, we lift the step limit and compare successful Screw Sorting episodes (Figure 6). SFEdit needs a median of 144 steps (mean 171), against 233 (mean 273) for the teacher and 204 (mean 244) for teleoperation, about 37% fewer than the teacher on average and faster than the human operators, as the mechanism of reinforcement learning boosts and favors faster completion; this comparison conditions on success. In Figure 7, RLT often fails to grasp at the height of the screw, and QAM hesitates, whereas SFEdit grasps the screw and aligns it with a free slot. The Plate Stacking failure marks the limit of action-level correction: the plate is placed on one whose color never appears in the demonstrations, a selection error of the frozen representation that no edit of the motion can repair. Human interventions, which prior systems use to correct such errors (Luo et al., 2025) during online reinforcement learning for robots, are excluded to measure autonomous learning alone.

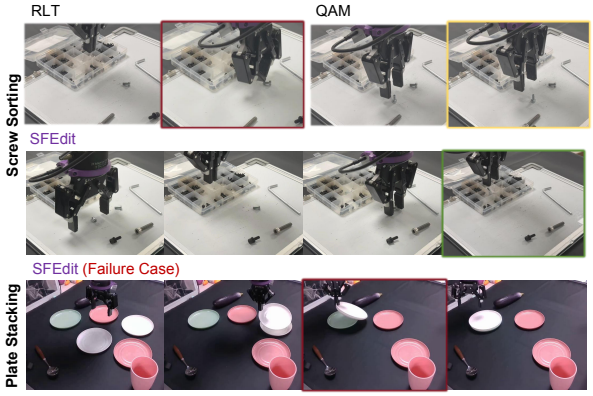

Figure 7: **Representative rollouts.** Framed images mark the decisive moment. Screw Sorting: RLT fails at the grasp (red), QAM hesitates (yellow), and SFEdit completes the grasp and placement (green). Plate Stacking: an SFEdit failure, stacking onto a plate whose color was absent from the demonstrations. The examples show where executions diverge, not how often.

6 Discussion

SFEdit treats generating an action and correcting it as one improvement problem with an explicit solution: the flow learns the envelope tilt, the executed policy is its proximal image, and one operator, built from the critic alone, serves learning, interaction, and bootstrapping. More broadly, the analysis suggests training a generator on the value its actions reach after they are corrected.

The theory assumes a fixed critic and exact optimization, which learned critics, approximate actors, and finite solves only approximate; an upper envelope can also spread an overestimated value peak, which a pessimistic critic would temper, and a movement penalty does not ensure safety. The frozen teacher representation also limits adaptation in unseen scenes, where the flow may propose only poor candidates that no local edit can repair. The transport paid off where corrections were large but not, on average, elsewhere; we apply it uniformly at all three sites, without tuning where it acts, so that SFEdit remains one operator rather than a tuned combination. The correction gains Δ come almost free with the transport: their mean is the net value of editing, and their spread governs the gap (8), so they could indicate in advance where editing and the envelope target pay off.

Reproducibility statement

Appendices A–F contain all proofs and analytic checks, and Figure 2 is an exact computation on a grid of 4,001 actions. Appendices G–K specify the implementation, baseline configurations, simulation protocol, ablation arms, and real-robot setup, with hyperparameters in Tables 1, 3, and 5. We will release code, configurations, per-run evaluation records, and the script for Figure 2.

AI use statement

We used large language models to polish text in all sections and to search for related work. They assisted with the theory part by checking proofs in Appendices A–F, including the short-edit and regularization-cost analyses of Appendix C.4, which they proposed during revision. They also wrote the script for the one-dimensional example of Figure 2, which the authors analyze the results. The authors conceived the method, implemented it, designed and ran all simulation and real-robot experiments, and take full responsibility for the paper.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1806.06920.

Charalambos D. Aliprantis and Kim C. Border. *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Springer, 3rd edition, 2006. doi: 10.1007/3-540-29587-9. URL https://link.springer.com/book/10.1007/3-540-29587-9.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Birkhäuser, 2nd edition, 2008. doi: 10.1007/978-3-7643-8722-8. URL https://link.springer.com/book/10.1007/978-3-7643-8722-8.

Ananth Balashankar, Ziteng Sun, Jonathan Berant, Jacob Eisenstein, Michael Collins, Adrian Hutter, Jong Lee, Chirag Nagpal, Flavien Prost, Aradhana Sinha, Ananda Theertha Suresh, and Ahmad Beirami. InfAlign: Inference-aware language model alignment. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 2646–2672, 2025. URL https://proceedings.mlr.press/v267/balashankar25a.html.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1577–1594, 2023. URL https://proceedings.mlr.press/v202/ball23a.html.

Ricardo Baptista, Panagiota Birmpa, Markos A. Katsoulakis, Luc Rey-Bellet, and Benjamin J. Zhang. Proximal optimal transport divergences. *Quarterly of Applied Mathematics*, 2026. doi: 10.1090/qam/1749. URL https://doi.org/10.1090/qam/1749.

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 Conference on Robot Learning (CoRL)*, pp. 17–40, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025b. URL https://arxiv.org/abs/2410.24164.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://roboticsproceedings.org/rss19/p026.html.

John M. Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966. doi: 10.1137/0114053. URL https://epubs.siam.org/doi/10.1137/0114053.

Carles Domingo-Enrich and Jiequn Han. Adjoint matching through the lens of the stochastic maximum principle in optimal control. *arXiv preprint arXiv:2604.08580*, 2026. URL https://arxiv.org/abs/2604.08580.

Carles Domingo-Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky T. Q. Chen. Stochastic optimal control matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2312.02027.

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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2409.08861.

Carles Domingo-Enrich, Yuanqi Du, and Michael S. Albergo. A unified perspective on fine-tuning and sampling with diffusion and flow models. *arXiv preprint arXiv:2605.00229*, 2026. URL https://arxiv.org/abs/2605.00229.

Perry Dong, Qiyang Li, Dorsa Sadigh, and Chelsea Finn. EXPO: Stable reinforcement learning with expressive policies. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026a. URL https://openreview.net/forum?id=aFjSjkB6CV.

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

Paul Dupuis and Richard S. Ellis. *A Weak Convergence Approach to the Theory of Large Deviations*. Wiley, 1997. doi: 10.1002/9781118165904. URL https://onlinelibrary.wiley.com/doi/book/10.1002/9781118165904.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.12557.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.13447.

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

Aaron J. Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Daniel S. Levine, Brandon M. Wood, Bin Hu, Brandon Amos, Brian Karrer, Xiang Fu, Guan-Horng Liu, and Ricky T. Q. Chen. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 22204–22237, 2025. URL https://proceedings.mlr.press/v267/havens25a.html.

Hokyun Im, Andrey Kolobov, Jianlong Fu, and Youngwoon Lee. Latent policy steering through one-step flow policies. In *Proceedings of Robotics: Science and Systems (RSS)*, 2026. URL https://arxiv.org/abs/2603.05296.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the Fokker–Planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998. doi: 10.1137/S0036141096303359. URL https://epubs.siam.org/doi/10.1137/S0036141096303359.

Qiyang Li and Sergey Levine. Q-learning with adjoint matching. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2601.14234.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2507.07969.

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

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.00927.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 2025. doi: 10.1126/scirobotics.ads5033. URL https://doi.org/10.1126/scirobotics.ads5033.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2507.21053.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965. doi: 10.24033/bsmf.1625. URL https://www.numdam.org/item/BSMF_1965__93__273_0/.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In *Proceedings of the Conference on Robot Learning (CoRL)*, pp. 4996–5013, 2024. URL https://proceedings.mlr.press/v270/nakamoto25a.html.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025a. URL https://arxiv.org/abs/2410.20092.

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025b. URL https://arxiv.org/abs/2502.02538.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1607–1612, 2010. doi: 10.1609/aaai.v24i1.7727. URL https://ojs.aaai.org/index.php/AAAI/article/view/7727.

David Pfau, Ian Davies, Diana L. Borsa, João Guilherme Madeira Araújo, Brendan Daniel Tracey, and Hado van Hasselt. Wasserstein policy optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 49128–49149, 2025. URL https://proceedings.mlr.press/v267/pfau25a.html.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S. Albergo. Tilt matching for scalable sampling and fine-tuning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2512.21829.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via Q-score matching. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 41163–41182, 2024. URL https://proceedings.mlr.press/v235/psenka24a.html.

Allen Z. Ren, Justin Lidard, Lars L. Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2409.00588.

R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1998. doi: 10.1007/978-3-642-02431-3. URL https://link.springer.com/book/10.1007/978-3-642-02431-3.

Max Sobol Mark, Tian Gao, Georgia Gabriela Sampaio, Mohan Kumar Srirama, Archit Sharma, Chelsea Finn, and Aviral Kumar. Policy agnostic RL: Offline RL and online RL fine-tuning of any class and backbone. *arXiv preprint arXiv:2412.06685*, 2024. URL https://arxiv.org/abs/2412.06685.

Andrew Wagenmaker, Yunchu Zhang, Mitsuhiko Nakamoto, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, pp. 258–282, 2025. URL https://proceedings.mlr.press/v305/wagenmaker25a.html.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2208.06193.

Charles Xu, Jost Tobias Springenberg, Michael Equi, Ali Amin, Adnan Esmail, Sergey Levine, and Liyiming Ke. RL token: Bootstrapping online RL with vision-language-action models. *arXiv preprint arXiv:2604.23073*, 2026a. URL https://arxiv.org/abs/2604.23073.

Haoran Xu, Kaiwen Hu, Somayeh Sojoudi, and Amy Zhang. Reinforcement learning via value gradient flow. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026b. URL https://arxiv.org/abs/2604.14265.

Long Yang, Zhixiong Huang, Fenghao Lei, Yucun Zhong, Yiming Yang, Cong Fang, Shiting Wen, Binbin Zhou, and Zhouchen Lin. Policy representation via diffusion probability model for reinforcement learning. *arXiv preprint arXiv:2305.13122*, 2023. URL https://arxiv.org/abs/2305.13122.

Guojian Zhan, Letian Tao, Pengcheng Wang, Yixiao Wang, Yiheng Li, Yuxin Chen, Hongyang Li, Masayoshi Tomizuka, and Shengbo Eben Li. Mean flow policy with instantaneous velocity constraint for one-step action generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2602.13810.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2204.13902.

Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. ReinFlow: Fine-tuning flow matching policy with online reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.22094.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.016. URL https://roboticsproceedings.org/rss19/p016.html.

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

A VARIATIONAL FORMULATION AND EXACT FACTORIZATION

This appendix treats one state and one fixed critic. Let $\mathcal{A} \subset \mathbb{R}^d$ be nonempty and compact, $\pi_{\mathrm{ref}} \in \mathcal{P}(\mathcal{A})$, and $Q : \mathcal{A} \to \mathbb{R}$ continuous. Write

$$g(a,b) = \beta Q(b) - \frac{\|b-a\|^2}{2\kappa}, \qquad U_\kappa(a) = \max_{b\in\mathcal{A}} g(a,b), \qquad Z = \int e^{U_\kappa(a)}\, d\pi_{\mathrm{ref}}(a). \tag{13}$$

Here $\mathcal{P}(\mathcal{A})$ denotes probability measures on $\mathcal{A}$. Measure notation permits atoms and lower-dimensional reference support; no Lebesgue density is required. The smooth adjoint implementation needs additional assumptions, stated in Appendix D.

A.1 EXISTENCE AND PROPERTIES OF THE REGULARIZER

Eliminating the intermediate law in (5) gives

$$\mathrm{KL}_\kappa(\pi\|\pi_{\mathrm{ref}}) = \inf_{\mu\in\mathcal{P}(\mathcal{A})} \left\{ \mathrm{KL}(\mu\|\pi_{\mathrm{ref}}) + \frac{W_2^2(\mu,\pi)}{2\kappa} \right\}. \tag{14}$$

Continuity and compactness imply that each maximum in (13) is attained. The measurable maximum theorem (Aliprantis & Border, 2006) gives continuity of U_κ and a nonempty compact maximizing correspondence with a measurable selection T_κ. Boundedness of U_κ gives $0 < Z < \infty$.

For fixed π, the infimum in (14) is attained. Indeed, $\mathcal{P}(\mathcal{A})$ is weakly compact, relative entropy is weakly lower semicontinuous, and W_2 metrizes weak convergence on compact $\mathcal{A}$. Taking $\mu = \pi_{\mathrm{ref}}$ gives a finite candidate. Consequently,

$$0 \le \mathrm{KL}_\kappa(\pi\|\pi_{\mathrm{ref}}) \le \min\left\{ \mathrm{KL}(\pi\|\pi_{\mathrm{ref}}), \frac{W_2^2(\pi,\pi_{\mathrm{ref}})}{2\kappa} \right\}. \tag{15}$$

Moreover $\mathrm{KL}_\kappa(\pi\|\pi_{\mathrm{ref}}) = 0$ if and only if $\pi = \pi_{\mathrm{ref}}$: an attaining measure must make both nonnegative terms vanish. This implication uses attainment of the infimum.

Following proximal optimal-transport regularization (Baptista et al., 2026), the construction is the Moreau–Yosida envelope of the functional $\mu \mapsto \mathrm{KL}(\mu\|\pi_{\mathrm{ref}})$ in the W_2 metric. Its inner minimization has the form of a JKO proximal step. SFEdit uses the *value* of this minimization as a regularizer in a separate reward-maximization problem; it does not identify the complete SFEdit update with the classical Fokker–Planck JKO evolution. Because (15) remains finite when π is not absolutely continuous with respect to π_{ref}, the regularizer allows spatial movement outside the reference support at finite cost.

A.2 PROOF OF THEOREM 1

For probability measures μ, π, let $\Pi(\mu,\pi)$ be their couplings. By the Kantorovich definition,

$$W_2^2(\mu,\pi) = \min_{\Gamma\in\Pi(\mu,\pi)} \int \|b-a\|^2\, d\Gamma(a,b). \tag{16}$$

The second marginal is unconstrained in the joint maximization. Substituting the coupling representation into (5) therefore yields

$$\sup_{\mu,\pi} \left\{ \beta\mathbb{E}_\pi Q - \mathrm{KL}(\mu\|\pi_{\mathrm{ref}}) - \frac{W_2^2(\mu,\pi)}{2\kappa} \right\} = \sup_{\mu} \left\{ -\mathrm{KL}(\mu\|\pi_{\mathrm{ref}}) + \sup_{\Gamma:\Gamma_1=\mu} \int g(a,b)\, d\Gamma(a,b) \right\}. \tag{17}$$

Only suprema are combined; no minimax exchange is required.

Disintegrate $\Gamma(da,db) = \mu(da)\Gamma_a(db)$. For every anchor a,

$$\int g(a,b)\, d\Gamma_a(b) \le U_\kappa(a). \tag{18}$$

The deterministic kernel concentrated at $T_\kappa(a)$ attains equality. Hence the right-hand side of (17) is

$$\sup_{\mu} \left\{ \mathbb{E}_\mu U_\kappa - \mathrm{KL}(\mu\|\pi_{\mathrm{ref}}) \right\}. \tag{19}$$

The Gibbs variational formula (Dupuis & Ellis, 1997) can be verified directly. Define $d\mu^+ = Z^{-1}e^{U_\kappa}d\pi_{\mathrm{ref}}$. For every μ of finite relative entropy,

$$\mathrm{KL}(\mu\|\mu^+) = \mathrm{KL}(\mu\|\pi_{\mathrm{ref}}) - \mathbb{E}_\mu U_\kappa + \log Z. \tag{20}$$

It follows that (19) equals $\log Z$, uniquely attained at μ^+.

Let $\Gamma^+ = (\mathrm{id}, T_\kappa)_\# \mu^+$ and $\pi^+ = (T_\kappa)_\# \mu^+$. This pair and coupling attain $\log Z$ in the coupling problem. Their joint objective in (5) is at least this value because $W_2^2(\mu^+, \pi^+)$ cannot exceed the cost of Γ^+. It also cannot exceed the global supremum $\log Z$. Thus equality holds, proving both attainment and the factorization. This also proves that Γ^+ is an optimal quadratic-cost coupling for its marginals. □

When proximal maximizers are not unique, an optimal conditional kernel may randomize among them. Every deterministic measurable selection still gives an optimizer. The intermediate law is unique, but the final law and its transport need not be. These distinctions are useful when a critic has several equal maxima.

A.3 Reference regularization and movement scales

Dividing (5) by β gives

$$\mathbb{E}_\pi Q - \frac{1}{\beta}\,\mathrm{KL}(\mu\|\pi_{\mathrm{ref}}) - \frac{1}{2\kappa\beta}W_2^2(\mu,\pi). \tag{21}$$

Thus β and κ provide two adjustable coefficients, but they are not orthogonal controls: changing β also changes the relative movement penalty. Rescaling Q requires accounting for the product $\kappa\beta$ in the proximal map. A fixed κ specifies a penalty, not an exact displacement radius or a cumulative trust-region budget.

B Envelope derivatives, transport, and improvement

B.1 Proof of Lemma 3 and boundary conditions

Extend U_κ to anchors in $\mathbb{R}^d$ using the same maximization over $b \in \mathcal{A}$. Expanding the quadratic gives

$$U_\kappa(a) + \frac{\|a\|^2}{2\kappa} = \max_{b\in\mathcal{A}} \left\{ \beta Q(b) - \frac{\|b\|^2}{2\kappa} + \frac{\langle a, b\rangle}{\kappa} \right\}. \tag{22}$$

The right-hand side is finite and convex, with locally bounded slopes because $\mathcal{A}$ is compact. Hence U_κ is locally Lipschitz and differentiable Lebesgue-almost everywhere. The subdifferential of the convex function in (22) is the convex hull of b/κ over maximizing b. At a unique maximizer, subtracting the quadratic gradient yields (9). Conversely, distinct maximizers give distinct slopes, so differentiability fails there. Almost-everywhere differentiability under Lebesgue measure does not automatically imply almost-sure differentiability for a reference law with atoms; uniqueness is the sufficient condition used by the smooth actor derivation.

Suppose now that $\mathcal{A}$ is convex and Q is continuously differentiable on a neighborhood of $\mathcal{A}$. At every global maximizer $b^* = T_\kappa(a)$,

$$\left\langle \beta\nabla Q(b^*) - \frac{b^* - a}{\kappa},\, b - b^* \right\rangle \le 0, \qquad b \in \mathcal{A}. \tag{23}$$

For a closed convex set, let $P_{\mathcal{A}}(z)$ denote its Euclidean nearest point,

$$P_{\mathcal{A}}(z) = \arg\min_{b\in\mathcal{A}} \|b - z\|^2. \tag{24}$$

This point is unique, and $P_{\mathcal{A}}$ is nonexpansive (Rockafellar & Wets, 1998). On a box, it is coordinatewise clipping. With the outward normal cone $N_{\mathcal{A}}(b^*) = \{w : \langle w, b - b^*\rangle \le 0 \text{ for all } b \in \mathcal{A}\}$, condition (23) is equivalent to

$$\beta\nabla Q(b^*) - \frac{b^* - a}{\kappa} \in N_{\mathcal{A}}(b^*), \qquad b^* = P_{\mathcal{A}}\big(a + \kappa\beta\nabla Q(b^*)\big). \tag{25}$$

At an interior optimum the normal cone vanishes, so $\nabla U_\kappa(a) = \beta\nabla Q(b^*)$. At a boundary optimum this equality generally fails; the displacement is the correct envelope gradient. For a nonconcave proximal objective, a fixed point is only a stationary point and need not be a global maximizer.

B.2 Proof of Proposition 4

By proximal optimality and feasibility of $b = a$,

$$\beta Q(T_\kappa(a)) = U_\kappa(a) + \frac{\|T_\kappa(a) - a\|^2}{2\kappa}, \qquad U_\kappa(a) \geq \beta Q(a). \tag{26}$$

The Gibbs identity (20) at $\mu = \mu^+$ implies

$$\mathbb{E}_{\mu^+} U_\kappa = \log Z + \mathrm{KL}(\mu^+ \| \pi_{\mathrm{ref}}). \tag{27}$$

Jensen's inequality gives $\log Z \geq \mathbb{E}_{\pi_{\mathrm{ref}}} U_\kappa \geq \beta \mathbb{E}_{\pi_{\mathrm{ref}}} Q$. Taking expectations in (26) therefore proves (10).

For the displacement bound, set $b = a$ in (23) and apply Cauchy–Schwarz:

$$\frac{\|b^* - a\|^2}{\kappa} \leq \beta \langle \nabla Q(b^*), b^* - a \rangle \leq \beta G \|b^* - a\|. \tag{28}$$

Dividing by the nonzero displacement proves the claim; a zero displacement satisfies it directly. The same bound holds for every fixed-point iterate after the initial anchor. This bounds movement when a gradient bound is available; it does not make κ alone a hard action radius. □

Equation (26) also gives the pointwise statement

$$Q(T_\kappa(a)) - Q(a) \geq \frac{\|T_\kappa(a) - a\|^2}{2\kappa\beta}. \tag{29}$$

This is the appropriate exact-solve comparison for an anchor–endpoint diagnostic. The distributional bound (10) compares different laws and should not be applied to a single action pair as though its KL term were pointwise.

B.3 Limiting behavior

Let $D = \mathrm{diam}(\mathcal{A})$ and $B_Q = \mathrm{osc}_{\mathcal{A}}(\beta Q)$. For a maximizer, (26) implies

$$\|T_\kappa(a) - a\| \leq \sqrt{2\kappa B_Q}. \tag{30}$$

By uniform continuity of βQ, its modulus of continuity ω then gives

$$0 \leq U_\kappa(a) - \beta Q(a) \leq \omega(\sqrt{2\kappa B_Q}) \longrightarrow 0 \quad \text{as } \kappa \downarrow 0, \tag{31}$$

uniformly in a. Consequently the intermediate law approaches the ordinary βQ tilt in total variation, and the transport approaches the identity uniformly. This describes the small-movement limit of the objective.

As $\kappa \to \infty$,

$$\max_{\mathcal{A}} \beta Q - \frac{D^2}{2\kappa} \leq U_\kappa(a) \leq \max_{\mathcal{A}} \beta Q. \tag{32}$$

Thus $\mu^+ \to \pi_{\mathrm{ref}}$ in total variation and every accumulation point of $T_\kappa(a)$ maximizes Q. If there are several maximizers, this does not imply convergence to a unique map. Uniform convergence of the envelope in the small-κ limit also does not by itself prove convergence of its gradient. At boundary anchors, the gradient limit can differ from the raw critic gradient because of the action constraint.

B.4 What smoothing does and does not imply

Infimal-convolution regularization does not make the upper envelope a conservative critic, nor does it imply global differentiability. It can spread the influence of a narrow, overestimated peak. An interior calculation makes the distinction precise. At a unique interior maximizer, assume Q is twice differentiable and $I - \kappa\beta\nabla^2 Q(T_\kappa(a))$ is invertible. Differentiating the stationary condition gives

$$DT_\kappa(a) = \big(I - \kappa\beta\nabla^2 Q(T_\kappa(a))\big)^{-1}, \tag{33}$$

and hence

$$\nabla^2 U_\kappa(a) = \beta\nabla^2 Q(T_\kappa(a))\big(I - \kappa\beta\nabla^2 Q(T_\kappa(a))\big)^{-1}. \tag{34}$$

A negative curvature eigenvalue is attenuated in magnitude; a positive eigenvalue below $1/(\kappa\beta)$ can be amplified. SFEdit therefore provides an explicit geometry for improvement, not a universal bound on critic-induced instability or a guarantee that the tilt is always flatter.

C Approximate proximal solves and learned actors

C.1 Convergence and a residual bound

Assume $\mathcal{A}$ is compact and convex, $\|\nabla Q\| \le G$, and ∇Q is L-Lipschitz. Define

$$\Phi_a(y) = P_{\mathcal{A}}\big(a + \kappa\beta\nabla Q(y)\big), \qquad c = \kappa\beta L. \tag{35}$$

Nonexpansiveness of projection implies $\|\Phi_a(y) - \Phi_a(z)\| \le c\|y - z\|$. If $c < 1$, Banach's fixed-point theorem gives a unique fixed point. A global maximizer exists by compactness and is a fixed point by (25); therefore this unique point is $T_\kappa(a)$. Starting from $y_0 = a$,

$$\|y_M - T_\kappa(a)\| \le c^M \|a - T_\kappa(a)\| \le c^M \kappa\beta G. \tag{36}$$

For the residual $r_M = \|y_M - \Phi_a(y_M)\|$, the triangle inequality gives

$$\|y_M - T_\kappa(a)\| \le r_M + c\|y_M - T_\kappa(a)\| \quad \Longrightarrow \quad \|y_M - T_\kappa(a)\| \le \frac{r_M}{1-c}. \tag{37}$$

The corresponding terminal-score error is bounded by

$$\left\| \frac{y_M - a}{\kappa} - \nabla U_\kappa(a) \right\| \le \min\left\{ c^M \beta G, \frac{r_M}{\kappa(1-c)} \right\}. \tag{38}$$

The residual requires an additional critic gradient if evaluated after the final iteration. It certifies proximity to the global maximizer only under the contraction condition. Outside this regime, the residual measures violation of the fixed-point equation but does not bound distance to a global maximizer. The contraction is in the solver variable y with anchor a fixed; it is not a claim that $a \mapsto T_\kappa(a)$ is a contraction. The implementation uses the relaxed iteration $y_{j+1} = P_{\mathcal{A}}\big((1-\rho)y_j + \rho\,(a + \kappa\beta\nabla Q(y_j))\big)$ with $\rho = 1/2$ (Appendix G.3); it has the same fixed points, and because $P_{\mathcal{A}}$ is nonexpansive, the map $y \mapsto P_{\mathcal{A}}\big((1-\rho)y + \rho\,(a + \kappa\beta\nabla Q(y))\big)$ is Lipschitz with constant $(1-\rho) + \rho c = 1 - \rho(1-c)$, so the bounds above hold with c replaced by $1 - \rho(1-c)$ and the residual computed with the relaxed map.

C.2 The one-iteration case

With $M = 1$ and inactive clipping,

$$\widehat{T}_\kappa(a) = a + \kappa\beta\nabla Q(a), \qquad -\frac{\widehat{T}_\kappa(a) - a}{\kappa} = -\beta\nabla Q(a). \tag{39}$$

Thus the actor seed coincides with the unmodified critic seed; with the relaxed iteration of Appendix G.3 it is that seed scaled by ρ. The behavior and bootstrap actions are still transported. Isolating the effect of the envelope target therefore requires several iterations or a more accurate solve, which is why we use $M = 4$. With active clipping, the seed is the feasible displacement divided by $-\kappa$.

C.3 An exact decomposition of implementation error

The factorization admits a useful error accounting that separates the learned anchor law from the proximal solve. Let $\widehat{\mu} \ll \pi_{\mathrm{ref}}$ have finite relative entropy, let $\widehat{T} : \mathcal{A} \to \mathcal{A}$ be measurable, and set $\widehat{\pi} = \widehat{T}_{\#}\widehat{\mu}$. Define the pointwise proximal suboptimality

$$e(a) = U_\kappa(a) - \left[\beta Q(\widehat{T}(a)) - \frac{\|\widehat{T}(a) - a\|^2}{2\kappa} \right] \ge 0, \tag{40}$$

and the achieved joint value using this explicit coupling,

$$\mathcal{V}(\widehat{\mu}, \widehat{T}) = \beta\mathbb{E}_{\widehat{\pi}} Q - \mathrm{KL}(\widehat{\mu}\|\pi_{\mathrm{ref}}) - \frac{\mathbb{E}_{\widehat{\mu}}\|\widehat{T}(a) - a\|^2}{2\kappa}. \tag{41}$$

Substitution of (40) into (20) yields the identity

$$\log Z - \mathcal{V}(\widehat{\mu}, \widehat{T}) = \underbrace{\mathrm{KL}(\widehat{\mu}\|\mu^+)}_{\text{anchor-law error}} + \underbrace{\mathbb{E}_{\widehat{\mu}} e(a)}_{\text{proximal error}}. \tag{42}$$

Since the Wasserstein cost is no larger than the cost of the explicit coupling,

$$0 \le \log Z - [\beta \mathbb{E}_{\widehat{\pi}} Q - \mathrm{KL}_\kappa(\widehat{\pi} \| \pi_{\mathrm{ref}})] \le \mathrm{KL}(\widehat{\mu} \| \mu^+) + \mathbb{E}_{\widehat{\mu}} e. \tag{43}$$

This directly relates the actual executed law to the ideal endpoint objective. It also gives the approximate improvement bound

$$\beta\big(\mathbb{E}_{\widehat{\pi}} Q - \mathbb{E}_{\pi_{\mathrm{ref}}} Q\big) \ge \mathrm{KL}(\widehat{\mu} \| \pi_{\mathrm{ref}}) + \frac{\mathbb{E}_{\widehat{\mu}} \|\widehat{T}(a) - a\|^2}{2\kappa} - \mathrm{KL}(\widehat{\mu} \| \mu^+) - \mathbb{E}_{\widehat{\mu}} e. \tag{44}$$

Neither error term is assumed to vanish in neural training. Here $\widehat{\mu}$ is the sampler's probability law, not a finite-batch empirical measure. Its mismatch can include actor approximation and numerical sampling error. A small empirical adjoint loss does not by itself provide a bound on this KL. These expressions are analytical quantities, not extra KL estimators required by SFEdit.

For a numerical solver, the readily measured quantity is $g(a, \widehat{T}(a)) - \beta Q(a)$. A nonnegative value guarantees (29) for that returned endpoint. Fixed-point iteration does not guarantee this test for arbitrary M outside the contraction regime. A monotone inner optimizer initialized at the anchor could enforce this test, but is a distinct solver from Algorithm 1.

C.4 Why the ordinary tilt followed by transport is generally suboptimal

QAM considers joint KL and Wasserstein constraints and approximates the intermediate law by the ordinary critic tilt before fitting a subsequent editor (Li & Levine, 2026). We analyze the corresponding quadratically penalized objective with an *exact* proximal editor, separating the choice of anchor law from editor approximation.

Let $d\mu_Q = Z_Q^{-1} e^{\beta Q} d\pi_{\mathrm{ref}}$ and define the available correction gain

$$\Delta(a) = U_\kappa(a) - \beta Q(a) \ge 0. \tag{45}$$

Then $d\mu^+ = e^{\Delta} d\mu_Q / \mathbb{E}_{\mu_Q} e^{\Delta}$: the optimal anchor law reweights the ordinary tilt by the value available through a penalized correction. Substituting $\widehat{\mu} = \mu_Q$ and $\widehat{T} = T_\kappa$ in (42) gives

$$\log Z - \mathcal{V}(\mu_Q, T_\kappa) = \mathrm{KL}(\mu_Q \| \mu^+) = \log \mathbb{E}_{\mu_Q} e^{\Delta} - \mathbb{E}_{\mu_Q} \Delta. \tag{46}$$

All terms are finite because Q and U_κ are bounded. Strict convexity of the exponential implies that the gap is zero exactly when Δ is constant μ_Q-almost surely. For example, on $[-1, 1]$ with $Q(b) = -b^2/2$, the proximal map is $a/(1 + \kappa\beta)$ and

$$\Delta(a) = \frac{\kappa \beta^2 a^2}{2(1 + \kappa\beta)}. \tag{47}$$

The gap is positive whenever a^2 is nonconstant under the reference. Conversely, a reference supported on $\{-c, c\}$ has zero gap despite the critic's nonzero curvature. Thus variation in correction gain, rather than curvature alone, determines strictness.

Short edits. If Q is twice continuously differentiable and $\kappa\beta\|\nabla^2 Q\| \ll 1$ near an interior anchor a, the stationary condition $b = a + \kappa\beta\nabla Q(b)$ gives $T_\kappa(a) - a = \kappa\beta\nabla Q(a) + O(\kappa^2)$ and

$$\Delta(a) = \tfrac{1}{2}\kappa\beta^2 \|\nabla Q(a)\|^2 + O(\kappa^2), \qquad \mathrm{KL}(\mu_Q \| \mu^+) = \tfrac{1}{8}\kappa^2\beta^4 \operatorname{Var}_{\mu_Q}\left(\|\nabla Q\|^2\right) + O(\kappa^3). \tag{48}$$

The envelope tilt thus reweights the critic tilt by $\exp(\kappa\beta^2\|\nabla Q\|^2/2)$, favoring anchors on steep ascent directions, and the gap is governed by how unevenly this slope is distributed across anchors. The expansion is in the dimensionless quantities $\kappa\beta\|\nabla Q\|$, relative to the size of $\mathcal{A}$, and $\kappa\beta\|\nabla^2 Q\|$, not in κ alone: the same κ gives short edits for a flat critic and long ones for a steep critic, whose action gradients scale with its returns.

The one-dimensional example. Figure 2 uses $\mathcal{A} = [-1, 1]$, the critic $Q(a) = 0.6\exp[-(a + 0.4)^2/(2 \cdot 0.2^2)] + \exp[-(a - 0.55)^2/(2 \cdot 0.04^2)]$, the reference $\pi_{\mathrm{ref}} \propto 0.55\mathcal{N}(-0.35, 0.3^2) + 0.45\mathcal{N}(0.25, 0.3^2)$ restricted to $\mathcal{A}$, $\beta = 3$, and $\kappa = 0.02$, all on a grid of 4,001 actions. Its gap is 0.27 nats. The envelope solution reaches the executed value $\mathbb{E}_{\pi^+} Q = 0.76$ at the regularization cost $\mathrm{KL}(\mu^+ \| \pi_{\mathrm{ref}}) + \mathbb{E}_{\mu^+} \|T_\kappa(a) - a\|^2/(2\kappa) = 0.35 + 0.29 = 0.64$ nats. With the same editor, the

critic tilt reaches this value only at inverse temperature 6.1, where its cost is $1.29 + 0.06 = 1.35$ nats. This reflects a general property. For fixed κ, Theorem 1 maximizes $\beta\mathbb{E}_\pi Q - R(\mu,\pi)$ with $R(\mu,\pi) = \mathrm{KL}(\mu\|\pi_{\mathrm{ref}}) + W_2^2(\mu,\pi)/(2\kappa)$ for every β, so no pair with $R(\mu,\pi) \le R(\mu^+,\pi^+)$ attains a larger executed value. Comparisons at equal β are therefore comparisons at equal prices, and a critic tilt of any temperature, followed by any editor, cannot reach the executed value of the envelope solution at a lower regularization cost.

For completeness, $(\mathrm{id}, T_\kappa)_{\#}\mu$ is an optimal transport coupling for *any* anchor law μ. For a competing coupling with the same marginals, the pointwise inequality $g(a,b) \le U_\kappa(a)$ bounds its expected penalized value by $\mathbb{E}_\mu U_\kappa$, with equality for the proximal coupling. Since the endpoint value expectation is fixed by the second marginal, this proves minimal transport cost. Hence (46) is the exact gap for the joint objective at the pair $(\mu_Q, (T_\kappa)_{\#}\mu_Q)$. It is only an upper bound on the reduced endpoint-objective gap in (43), which can optimize over a different intermediate law. Nor does it rank neural implementations or imply a strict online-return improvement; those depend on approximation and critic quality.

C.5 Sensitivity to critic value errors

For continuous critics Q and $\widetilde{Q}$ with $\|Q - \widetilde{Q}\|_\infty \le \epsilon$, write $U, \widetilde{U}$ for their envelopes and $Z, \widetilde{Z}$ for their normalizers. Taking maxima of functions whose values differ by at most $\beta\epsilon$ gives

$$\|U - \widetilde{U}\|_\infty \le \beta\epsilon, \qquad |\log Z - \log \widetilde{Z}| \le \beta\epsilon. \tag{49}$$

The corresponding optimal anchor laws satisfy

$$\left|\log \frac{d\mu^+}{d\widetilde{\mu}^+}(a)\right| \le 2\beta\epsilon \qquad \pi_{\mathrm{ref}}\text{-almost surely.} \tag{50}$$

This controls values and anchor weights, but not the continuity of the maximizing endpoint. Nearly tied maxima can switch under arbitrarily small value perturbations; gradient or transport stability requires additional regularity. The result does not remove the need to assess critic error on edited actions.

C.6 From critic improvement to return improvement

Assume bounded rewards and measurable policy kernels, and consider fixed-length chunks with $\bar{\gamma} = \gamma^H$. Suppose the reference is an actual policy whose true chunk action value is $Q^{\pi_{\mathrm{ref}}}$, and assume

$$\sup_{s,a} |Q(s,a) - Q^{\pi_{\mathrm{ref}}}(s,a)| \le \epsilon_Q. \tag{51}$$

Let $\mathcal{G}(s)$ denote the right-hand side of (44) divided by β. Comparing the two expectations of Q and $Q^{\pi_{\mathrm{ref}}}$ gives

$$\mathbb{E}_{a\sim\widehat{\pi}(\cdot|s)} Q^{\pi_{\mathrm{ref}}}(s,a) - V^{\pi_{\mathrm{ref}}}(s) \ge \mathcal{G}(s) - 2\epsilon_Q. \tag{52}$$

The discounted performance-difference identity then yields, from initial state s_0,

$$V^{\widehat{\pi}}(s_0) - V^{\pi_{\mathrm{ref}}}(s_0) \ge \frac{1}{1-\bar{\gamma}} \mathbb{E}_{s\sim d^{\widehat{\pi}}_{s_0}} [\mathcal{G}(s) - 2\epsilon_Q], \tag{53}$$

where $d^{\widehat{\pi}}_{s_0}$ is the normalized discounted decision-state occupancy. To see this, expand the reference advantage along a trajectory of $\widehat{\pi}$; consecutive reference values telescope, leaving the return difference. Substitute (52) and use the definition of $d^{\widehat{\pi}}_{s_0}$.

This conditional result does not establish monotonic return improvement for Algorithm 1. Its critic is trained toward the current edited behavior, and its reference may be a Polyak average rather than that behavior. Neither the required uniform critic bound nor exact evaluation of the reference is guaranteed. The main paper accordingly states fixed-critic improvement and assesses online return empirically.

D ADJOINT IMPLEMENTATION AND ENDPOINT COORDINATES

D.1 REFERENCE FIELD AND TRAINING-PATH LAW

For the independent Gaussian-to-data linear interpolation and its exact marginal flow, the memoryless representation uses drift $2v(x,t) - x/t$ and noise $\sqrt{2(1-t)/t}$ on $0 < t < 1$. In the adjoint learner, the forward controlled drift uses the current field v_θ, whereas the lean adjoint differentiates v_{ref}. These are separate roles. Sampling all training paths from the reference instead would be a different training rule, for which the current-policy adjoint argument cannot simply be reused.

For a target tilt e^f, the terminal loss in the associated control formulation is $-f$. In SFEdit, $f = U_\kappa$; therefore the terminal condition is the negative envelope gradient. Given a path, the backward recursion constructs a detached regression target for (4). No derivative is propagated through the realized solver states, through the proximal iterations, or through the reference parameters. Computing $\nabla_a Q$ still requires differentiation with respect to the action; detaching critic parameters does not remove that action gradient.

The exact tilt interpretation assumes that the reference field realizes the interpolation marginals, the memoryless process is well defined with its limiting initial law, and the admissible control class can attain the control optimum (Domingo-Enrich et al., 2025; Domingo-Enrich & Han, 2026). An arbitrary learned field or its Polyak average need not be an exact marginal flow. Likewise, its finite-grid SDE and ODE samplers need not have identical endpoint laws. These discrepancies belong to actor and sampling error; Theorem 1 is a distributional result independent of this implementation.

D.2 TIME GRID AND SINGULAR ENDPOINTS

The continuous coefficients are singular at $t = 0$ and the noise vanishes at $t = 1$. Neither endpoint is a valid place to substitute the raw quotient in (4). A numerical implementation therefore fixes a time grid, an initialization, an endpoint treatment, and a loss quadrature. We write $t_0 < \cdots < t_K$ and $\tau_i = t_i + \varepsilon > 0$ for the working adjoint denominator. A strictly positive lower time cutoff corresponds to $\varepsilon = 0$ on a truncated grid; a shifted denominator corresponds to $\varepsilon > 0$.

A shift in the denominator is a numerical modification; it does not make the modified SDE exactly marginal-preserving or memoryless. A discrete path update with a shifted drift reads

$$x_{i+1} = x_i + h_i \left(2v_\theta(x_i, t_i \mid s) - \frac{x_i}{\tau_i} \right) + \sigma_i \sqrt{h_i}\, \xi_i, \qquad \xi_i \sim \mathcal{N}(0, I). \tag{54}$$

Here σ_i is the actual implementation schedule, not a value inferred from τ_i. A positive lower cutoff also requires the appropriate initial law at that cutoff; simply drawing fresh standard Gaussian noise there is an additional approximation. The loss sums over evaluation sites with finite, strictly positive σ_i, using quadrature weights w_i:

$$\widehat{\mathcal{L}_{\mathrm{AM}}} = \sum_{i \in \mathcal{I}} w_i \left\| \frac{2(v_\theta(x_i, t_i \mid s) - v_{\mathrm{ref}}(x_i, t_i \mid s))}{\sigma_i} + \sigma_i \lambda_i \right\|^2. \tag{55}$$

The iQAM algebra below applies to the stated adjoint denominator, without asserting that this choice alone fixes forward sampling error.

D.3 DIRECT STUDENT OUTPUTS AND ACTION COORDINATES

The student flow directly generates complete actions. Let u be its raw endpoint and $a = C_s(u) \in \mathcal{A}$ the action after any fixed coordinate conversion or clipping. It does not predict an additive residual on a teacher action. For the reference endpoint law ν_{ref}, the action reference is $\pi_{\mathrm{ref}} = (C_s)_\# \nu_{\mathrm{ref}}$.

The action-space envelope tilt is realized by the raw-space law

$$d\nu^+(u) = Z^{-1} e^{U_\kappa(C_s(u))} d\nu_{\mathrm{ref}}(u). \tag{56}$$

Indeed, for any bounded measurable ψ,

$$\mathbb{E}_{\nu^+} \psi(C_s(u)) = Z^{-1} \int \psi(a) e^{U_\kappa(a)} d\pi_{\mathrm{ref}}(a) = \mathbb{E}_{\mu^+} \psi(a), \tag{57}$$

so $(C_s)_{\#}\nu^+ = \mu^+$ even if C_s is not invertible. At differentiable points with a unique proximal maximizer, the terminal seed is

$$\lambda_K = -DC_s(u_K)^\top \frac{T_\kappa(C_s(u_K)) - C_s(u_K)}{\kappa}. \tag{58}$$

The executed action is $T_\kappa(C_s(u_K))$, approximated using $\widehat{T}_\kappa$. With direct outputs in the chosen action coordinates and no further clipping, C_s is the identity and the main-text seed applies. Box clipping has zero derivative on saturated coordinates almost everywhere; a straight-through derivative changes this estimator. In simulation, the solver starts from the raw flow endpoint x_K and clips every iterate, so for an endpoint outside the box the terminal signal $-(\widehat{T}_\kappa(x_K) - x_K)/\kappa$ also contains the move onto the box; this is the envelope gradient of Lemma 3 with anchors in $\mathbb{R}^d$. QAM differentiates a critic whose action input is clipped, which gives saturated coordinates no signal, so comparisons between the two targets include this boundary term. A compressed observation representation need not be Markov, so the MDP analysis assumes that the conditioning contains sufficient state information.

E iQAM: DERIVATION AND SCOPE OF THE NUMERICAL IMPROVEMENT

Proposition 5 (Properties of iQAM)**.** *Fix a path and a grid $t_0 < \cdots < t_K$ with $\tau_i = t_i + \varepsilon > 0$. The iQAM update (11) is a right-endpoint approximation to the exact integral formula (62). If $\|J(t_{j+1})\| \le L_v$, its iterates satisfy*

$$\|\lambda_i\| \le \frac{\tau_i}{\tau_K} e^{2L_v(t_K - t_i)} \|\lambda_K\|. \tag{59}$$

If $J(t)^\top \lambda(t)$ is constant on one interval, iQAM's scalar quadrature weight is closer to the exact weight than direct Euler's. Both schemes have first-order global accuracy for time-Lipschitz J and a fixed positive lower bound on τ.

The following derivation proves each claim. The quadrature comparison is restricted to the stated constant-integrand case; it is not a uniform error ordering for learned fields or stochastic trajectories.

E.1 INTEGRATING FACTOR

Along a fixed path, set $J(t) = \nabla_x v_{\mathrm{ref}}(x_t, t \mid s)$ and $\tau(t) = t + \varepsilon > 0$. The working lean adjoint is

$$\dot\lambda(t) = -2J(t)^\top \lambda(t) + \frac{\lambda(t)}{\tau(t)}. \tag{60}$$

With $\lambda(t) = \tau(t) z(t)$ and $\dot\tau = 1$, the scalar terms cancel:

$$z + \tau \dot z = -2\tau J^\top z + z, \qquad \dot z = -2J^\top z. \tag{61}$$

Integrating backward from t_{i+1} to t_i and multiplying by τ_i gives

$$\lambda_i = \frac{\tau_i}{\tau_{i+1}} \lambda_{i+1} + 2 \int_{t_i}^{t_{i+1}} \frac{\tau_i}{\tau(t)} J(t)^\top \lambda(t)\, dt. \tag{62}$$

Applying the explicit right-endpoint rule to the integral in the z variable yields (11). This is explicit Euler after a change of variables, not the implicit method commonly called backward Euler.

Direct explicit Euler in λ instead gives

$$\lambda_i^{\mathrm{E}} = \left(1 - \frac{h_i}{\tau_{i+1}}\right) \lambda_{i+1} + 2h_i J_{i+1}^\top \lambda_{i+1} = \frac{\tau_i}{\tau_{i+1}} \lambda_{i+1} + 2h_i J_{i+1}^\top \lambda_{i+1}. \tag{63}$$

If $J = 0$, the exact solution is proportional to $\tau(t)$ and both formulas are exact. The difference is entirely in the learned-Jacobian increment. This is why removing the known term alone cannot establish a stability advantage over the original recursion.

E.2 Local analysis with a frozen Jacobian

If $J(t) = J$ is constant on an interval, the exact propagator is

$$\lambda_i = \frac{\tau_i}{\tau_{i+1}} e^{2h_i J^\top} \lambda_{i+1}. \tag{64}$$

iQAM replaces the matrix exponential by $I + 2h_i J^\top$, while direct Euler also omits the scalar factor on its linear Jacobian term. Expanding at fixed $\tau_{i+1} > 0$ shows that both schemes have local error of order h_i^2 and first-order global accuracy when the fixed-path coefficients are Lipschitz in time. iQAM removes one scalar–Jacobian cross-term from this frozen-Jacobian expansion; higher-order terms can have either sign, so this alone does not prove smaller error for every learned field.

For a scalar eigenvalue j of a constant Jacobian, the exact, direct-Euler, and iQAM multipliers are respectively

$$d_i e^{2h_i j}, \qquad d_i + 2h_i j, \qquad d_i(1 + 2h_i j), \qquad d_i = \tau_i/\tau_{i+1}. \tag{65}$$

The learned part remains explicitly discretized. Large $|h_i j|$ can still cause inaccurate or unstable propagation. iQAM does not eliminate the need to resolve rapidly varying learned dynamics.

E.3 A norm bound on arbitrary grids

The integrating factor also gives a simple stability bound that does not depend on resolving $1/\tau$ by small steps. Suppose $\|J(t_{j+1})\| \le L_v$ and $\tau_j > 0$. The iQAM update obeys

$$\begin{aligned} \|\lambda_i\| &\le \frac{\tau_i}{\tau_K} \prod_{j=i}^{K-1} (1 + 2h_j L_v) \, \|\lambda_K\| \\ &\le \frac{\tau_i}{\tau_K} e^{2L_v(t_K - t_i)} \|\lambda_K\|. \end{aligned} \tag{66}$$

Indeed, take operator norms in $\lambda_j = d_j(I + 2h_j J_{j+1}^\top)\lambda_{j+1}$, telescope the d_j, and use $1 + x \le e^x$. For bounded measurable $J(t)$, the continuous transformed equation has the same bound by Gronwall's inequality. Thus iQAM preserves this scalar attenuation even on a coarse grid. This is a bound on adjoint amplification, not a guarantee of small numerical error or better RL performance.

For accuracy, stronger temporal regularity matters. If J is Lipschitz along a fixed path and $\varepsilon > 0$ is fixed, Euler's method for z has a global $O(h_{\max})$ error, and multiplication by bounded τ preserves that order. A Jacobian evaluated along a stochastic trajectory need not be Lipschitz in time, so this deterministic order does not directly extend to the full SDE training procedure. Forward sampling error, stochastic quadrature, and a vanishing endpoint shift require their own analysis.

E.4 A precise quadrature-weight comparison

If $J(t)^\top \lambda(t)$ is constant on an interval, its exact scalar weight in (62) is

$$w_* = \tau_i \log\left(1 + \frac{h_i}{\tau_i}\right), \qquad w_{\mathrm{E}} = h_i, \qquad w_{\mathrm{I}} = \frac{h_i \tau_i}{\tau_{i+1}}. \tag{67}$$

Let $r = h_i/\tau_i > 0$. Then

$$\frac{w_{\mathrm{I}}}{h_i} = \frac{1}{1+r} \le \frac{w_*}{h_i} = \frac{\log(1+r)}{r} \le 1 = \frac{w_{\mathrm{E}}}{h_i}. \tag{68}$$

In this restricted comparison, iQAM is the closer weight in absolute error:

$$\frac{\log(1+r)}{r} - \frac{1}{1+r} \le 1 - \frac{\log(1+r)}{r}. \tag{69}$$

For a proof, define $\varphi(r) = r(2+r)/(2(1+r)) - \log(1+r)$. Then $\varphi(0) = 0$ and $\varphi'(r) = r^2/(2(1+r)^2) \ge 0$, which is equivalent to (69).

The normalization of an error matters. Normalized by h_i, the two absolute errors tend to 1 and 0, respectively, as $r \to \infty$. Relative to the *exact* weight, however, they are

$$\frac{w_{\mathrm{E}} - w_*}{w_*} = \frac{r}{\log(1+r)} - 1 \longrightarrow \infty, \qquad \frac{w_* - w_{\mathrm{I}}}{w_*} = 1 - \frac{r}{(1+r)\log(1+r)} \longrightarrow 1. \tag{70}$$

Both relative errors are $r/2 + O(r^2)$ as $r \downarrow 0$. This scalar calculation is not a general error bound when $J(t)^\top \lambda(t)$ varies.

On a uniform grid $t_i = ih$ with $\varepsilon = h$, the factor is $d_i = (i+1)/(i+2)$. The early factors therefore remain $1/2$ and $2/3$ as K changes, while the intervals themselves shrink. Standard fixed-ε consistency does not directly give a uniform theorem when $\varepsilon = h \to 0$. That singular limit needs separate analysis. Our numerical claim concerns the measured finite grids in the solver ablation. Because $d_i < 1$, iQAM also damps the Jacobian increment relative to direct Euler, most strongly on the early intervals: on this grid with $K = 4$, direct Euler weights it by 2, $3/2$, $4/3$, and $5/4$ times the iQAM weight. The weight is prescribed by the known scalar dynamics rather than tuned, and we did not compare it with arbitrary damping factors.

E.5 Cost and implementation

Both recursions require the same $J_{i+1}^\top \lambda_{i+1}$, evaluated as a VJP without forming the Jacobian. iQAM multiplies this increment by the precomputed scalar d_i. It adds no network evaluation or learned parameter. This statement is specific to the adjoint correction; the proximal solver of SFEdit adds critic-gradient evaluations, which are counted separately in Section 4.4.

F Analytic checks

A concave quadratic provides a closed-form check of the envelope, tilt, and transport. This subsection uses $\mathcal{A} = \mathbb{R}^d$, with finite-moment references, rather than the compact-domain assumptions of the main theorem. Let

$$Q(a) = q_0 - \tfrac{1}{2}(a-m)^\top A(a-m), \qquad A \succeq 0. \tag{71}$$

The proximal objective is strictly concave and its stationary point is

$$T_\kappa(a) = m + (I + \kappa\beta A)^{-1}(a-m), \tag{72}$$

with envelope

$$U_\kappa(a) = \beta q_0 - \frac{\beta}{2}(a-m)^\top A(I+\kappa\beta A)^{-1}(a-m). \tag{73}$$

Direct differentiation verifies (9). If the reference is Gaussian $\mathcal{N}(m_0, \Sigma_0)$ with positive-definite covariance, put $B = \beta A(I + \kappa\beta A)^{-1}$. The tilted law is Gaussian with

$$\Sigma_+ = (\Sigma_0^{-1} + B)^{-1}, \qquad m_+ = \Sigma_+(\Sigma_0^{-1} m_0 + Bm). \tag{74}$$

Its affine pushforward has mean $m + (I+\kappa\beta A)^{-1}(m_+ - m)$ and covariance $(I+\kappa\beta A)^{-1}\Sigma_+(I + \kappa\beta A)^{-1}$. These expressions verify the envelope seed and executed law without a neural critic. The normalizer also has the closed form

$$Z = e^{\beta q_0} \det(I + \Sigma_0 B)^{-1/2} \exp\left[-\tfrac{1}{2}(m_0 - m)^\top B(I+\Sigma_0 B)^{-1}(m_0 - m)\right]. \tag{75}$$

Here $B(I + \Sigma_0 B)^{-1}$ is symmetric, as follows by rewriting it as $(I + B\Sigma_0)^{-1}B$.

For a boundary check in one dimension, take $\mathcal{A} = [-1, 1]$ and $Q(b) = gb$ with $g > 0$. Then $T_\kappa(a) = \mathrm{clip}(a + \kappa\beta g, -1, 1)$. Near the upper boundary, the envelope gradient equals $(1-a)/\kappa$, rather than βg. This distinguishes the correct seed from the unconstrained endpoint gradient. A constant reference Jacobian gives the independent adjoint check (64).

G Simulation implementation

This section specifies the simulation implementation of SFEdit and its variants; the real-robot implementation is described in Appendix K.

G.1 Networks and optimization

Following QAM (Li & Levine, 2026), the actor is a single velocity field, parameterized by an MLP with four hidden layers of width 512, GELU activations, and no layer normalization. The critic is an ensemble of $N = 10$ MLPs of the same widths with layer normalization after each hidden activation.

All layers use variance-scaling initialization (scale 1, fan-average, uniform). One Adam optimizer with learning rate 3×10^{-4} updates the critics and the actor after global-norm gradient clipping at 1.0; in SFEdit, non-finite gradient entries are additionally set to zero, a guard that the baseline implementations do not have. Target critics are Polyak averages with rate 0.005, and the terminal adjoint uses their mean $Q = N^{-1} \sum_n Q_{\bar{\phi}_n}$, with the action input of the critic clipped to $[-1, 1]$. The batch size is 256, one gradient update is made per environment step, and all runs use action chunking with the per-domain horizon of Table 3.

G.2 Actor loss and reference

The actor minimizes $\mathcal{L}_{\mathrm{FM}} + c_{\mathrm{AM}} \mathcal{L}_{\mathrm{AM}}$ with $c_{\mathrm{AM}} = 1$, where the flow-matching term uses replay samples; with $c_{\mathrm{AM}} = 0$ the actor reduces to behavior cloning. The reference field v_{ref} is held fixed while the adjoint targets of a batch are built and is updated together with the target critics. The controlled field samples actions with $K = 10$ Euler steps in all main experiments; the four-step ablation changes nothing else.

G.3 Proximal solver

The transport is the relaxed projected fixed-point iteration

$$y_0 = a, \qquad y_{j+1} = \mathrm{clip}\left(y_j + \rho_{\mathrm{prox}}\, \kappa \Big(\beta \nabla_a Q(s, y_j) - \frac{y_j - a}{\kappa}\Big)\right), \qquad \widehat{T}_\kappa(a) = y_M, \tag{76}$$

which equals $\mathrm{clip}\left((1 - \rho_{\mathrm{prox}}) y_j + \rho_{\mathrm{prox}}(a + \kappa\beta \nabla_a Q(s, y_j))\right)$, with $M = 4$ iterations, relaxation $\rho_{\mathrm{prox}} = 0.5$, the anchor a held fixed, every iterate detached, and clipping to the action box; for $\kappa < 10^{-8}$ the transport is the identity. The relaxed iteration has the same fixed points as (12): by the projection characterization, $y = \mathrm{clip}(y + \rho(w - y))$ holds if and only if $\langle w - y, b - y \rangle \le 0$ for all $b \in \mathcal{A}$, which does not depend on $\rho > 0$, so a fixed point of (76) is exactly a solution of the first-order condition (25). Under the contraction condition of Appendix C.1, the relaxed map contracts with factor $1 - \rho_{\mathrm{prox}}(1 - \kappa\beta L)$. With $M = 1$ and inactive clipping, $y_1 - a = \rho_{\mathrm{prox}} \kappa \beta \nabla Q(a)$, so the one-iteration actor target is the raw critic gradient scaled by ρ_{prox} (Appendix C.2). Every SFEdit run applies the transport both to the executed action and to the bootstrap action.

G.4 Critic targets, chunking, and replay

An action is a chunk of H primitive commands executed open loop; the replay stores one transition per chunk with the chunk return $R = \sum_{j=0}^{H-1} \gamma^j r_j$ and the bootstrap discount γ^H. The 500k online budget counts environment steps, so the policy is queried $500\mathrm{k}/H$ times. The backup uses the ensemble mean minus ρ_{pes} times the ensemble standard deviation at the transported next action, with the pessimism coefficient ρ_{pes} of the QAM protocol (Table 3); all backup quantities are detached. A single replay buffer, preloaded with the offline dataset, receives all online transitions and is sampled uniformly. Updates start after 5,000 environment steps, and the warm-started policy acts from the first step without a separate exploration phase.

G.5 Hyperparameters

The inverse temperature $\beta = 3$ is shared by SFEdit, SFEdit(QAM), iQAM, and QAM in all ten domains and was not tuned per domain. The original QAM protocol instead tunes the temperature per domain (10 for antmaze-large, 1 for cube-double, cube-quadruple, and scene, 30 for puzzle-4×4, and 3 elsewhere); we compare all tilt-based methods at one temperature so that their differences reflect the target and the discretization rather than temperature tuning. The shared value is QAM's own in five domains, including antmaze-giant, which contributes 4.2 of the 6.9-point aggregate gain of SFEdit over QAM (Table 4); cube-quadruple and puzzle-4×4, which contribute 2.0 points each, use 1 and 30 in QAM's protocol, so QAM may be below its tuned setting there. SFEdit adds one hyperparameter, κ, set to 0.02 in nine domains and to 0.2 in antmaze-giant, which prefers a larger scale (Appendix J); FQL, DSRL, and EXPO use their published per-domain settings (Appendix H.3).

Table 1: Settings of the four tilt-based arms in simulation.

Quantity	Value	Note
Inverse temperature β	3.0 (all domains)	shared by SFEdit, SFEdit(QAM), iQAM, QAM
Movement scale κ	0.02; antmaze-giant 0.2	SFEdit arms only
Solver iterations M	4	anchor fixed
Relaxation ρ_{prox}	0.5	effective step $\rho_{\text{prox}}\kappa$
Flow steps K	10	4 in the step ablation
Transport sites	execution and backup	SFEdit arms only
Adjoint coefficient c_{AM}	1.0	0 gives the behavior-cloning actor
Adjoint recursion	integrating factor	SFEdit and iQAM; original in SFEdit(QAM) and QAM
Candidate count	1	no best-of-n selection
Batch, learning rate, clipping	256, 3×10^{-4}, 1.0	shared by all methods

H Baseline implementations

H.1 Sources

Every simulation baseline except QSM runs unchanged method code from a public repository: QAM, BAM, IFQL, RLPD, FQL, DSRL, and TRQAM from the QAM repository (Li & Levine, 2026), and EXPO from the EXPO repository (Dong et al., 2026a). QSM is re-implemented from Psenka et al. (2024) for our rectified-flow policy through the identity $\hat{\epsilon} = x_t - t\,v$, and we verified that it matches the reference formulation numerically.

H.2 Shared settings

All nine simulation baselines follow the protocol of the SFEdit arms: no offline steps, 500k online environment steps, updates from step 5,000, one update per step, evaluation every 50k steps on 50 episodes with the shared evaluation seed, learning rate 3×10^{-4}, batch 256, the architectures of Appendix G.1, Polyak rate 0.005, and ten flow steps where a flow is used. Every arm of a task is initialized from the same offline QAM checkpoint, loading the network modules that its architecture shares with that checkpoint. Each baseline contributes 50 tasks with five seeds, that is, 250 learning curves.

H.3 Method-specific settings

BAM. Basic adjoint matching uses $\beta = 3$ and ten flow steps, with all other settings at their defaults.

IFQL. Expectile 0.9 and 32 candidate samples, without further tuning.

RLPD. Ten critics, no behavior-cloning term, and an automatic entropy temperature initialized at 1.0, with the target entropy set to half the action dimension, as in the reference implementation.

TRQAM. The fixed β is replaced by a dual variable with scale 3.0 and initial value 1.0, which reproduces $\beta = 3$ at initialization, with KL budget 1.0, dual step 0.01, KL moving-average coefficient 0.1, and the dual variable clipped to $[0.01, 100]$.

QSM. The gradient is taken at the iterate as in the released code, the exploration noise is 0.1 (0 for all other flow methods), and the sampler is clipped, without which all runs score zero; the number of critic-gradient steps is 30, 10, or 3, depending on the compute server.

FQL, DSRL, and EXPO. These methods use the published per-domain values of their original implementations; in antmaze-giant and humanoidmaze-medium, for example, FQL uses $\alpha = 3.0$ and 30.0, DSRL a noise scale of 1.2 and 0.6, and EXPO an edit scale of 0.3 and 0.5.

H.4 Real-robot comparisons

The teacher is the frozen $\pi_{0.5}$ checkpoint of Appendix K. DSRL trains a soft actor–critic over the teacher's initial flow noise: a 32-dimensional latent action, broadcast over the ten chunk steps and scaled by $1.6\tanh(\cdot)$, conditioned on the 7-dimensional pose and gripper state and a 2,048-dimensional $\pi_{0.5}$ prefix feature. RLT conditions an actor on the same state and on the teacher's reference action chunk and outputs a complete 70-dimensional chunk, with a behavior-cloning weight of 25, reference dropout 0.5, an anchor radius of 0.5, a fixed exploration standard deviation of 0.02, and two critic updates per actor update (Xu et al., 2026a). QAM uses the student of SFEdit with the raw critic terminal signal, the original adjoint discretization, and no transport. Table 5 lists the remaining settings.

I Simulation protocol

I.1 Domains, tasks, and datasets

Table 2 lists the ten domains; each contributes five single-task environments, giving 50 tasks, and each task is run with five seeds. The five tasks of a domain share one dataset and differ only in reward relabeling. The experiments ran on servers with four RTX PRO 6000 Blackwell GPUs and six RTX A6000 GPUs; the main comparison comprises 3,000 runs (12 methods, 50 tasks, five seeds) and the ablations about 2,750 further runs, each of 500k environment steps, in addition to 50 offline pretraining runs of one million updates.

Table 2: Domains, environment names, and dataset settings. Task identifiers follow the pattern `{prefix}-singletask-task{1..5}-v0`.

Domain	Environment prefix	Sparse reward	Dataset
antmaze-giant	`antmaze-giant-navigate`	no	standard
antmaze-large	`antmaze-large-navigate`	no	standard
humanoidmaze-medium	`humanoidmaze-medium-navigate`	no	standard
humanoidmaze-large	`humanoidmaze-large-navigate`	no	standard
cube-double	`cube-double-play`	no	standard
cube-triple	`cube-triple-play`	no	standard
cube-quadruple	`cube-quadruple-play`	no	100M
puzzle-3×3	`puzzle-3x3-play`	yes	standard
puzzle-4×4	`puzzle-4x4-play`	no	100M
scene	`scene-play`	yes	standard

I.2 Offline pretraining and weak starts

Each task has one starting checkpoint, produced by offline QAM with one million offline updates and no online steps under a single pretraining seed, and all five online seeds of every method start from that file. The checkpoints come from QAM rather than SFEdit so that no baseline inherits our pretraining. The pretraining recipe uses batch 256, learning rate 3×10^{-4}, the architectures of Appendix G.1, ten critics, Polyak rate 0.005, ten flow steps, gradient clipping, and the per-domain discount, pessimism, and horizon of Table 3. The starting checkpoint is the latest checkpoint of the pretraining run whose measured success, on 50 episodes at the evaluation seed, does not exceed 0.10. Domain-level starting success under the QAM arm ranges from 0.000 (antmaze-large) to 0.036 (antmaze-giant, humanoidmaze-medium), and the mean over the twelve methods and 50 tasks is 0.023.

I.3 Online training

The common settings are those of Appendix G.4: 500k environment steps, no offline steps, updates from step 5,000, one update per step, batch 256, Adam at 3×10^{-4} with global-norm clipping 1.0, one uniformly sampled replay buffer preloaded with the offline data, and Polyak targets with rate 0.005. The discount, chunk horizon, pessimism coefficient, and movement scale differ by domain (Table 3); the discount and pessimism values follow the QAM protocol.

Table 3: Per-domain training settings. The discount and pessimism coefficient follow the QAM protocol; the movement scale applies to the SFEdit arms.

Domain	Discount γ	Horizon H	Pessimism ρ_{pes}	κ
antmaze-giant	0.995	1	0.5	0.2
antmaze-large	0.995	1	0.5	0.02
humanoidmaze-medium	0.995	1	0.0	0.02
humanoidmaze-large	0.999	1	0.0	0.02
cube-triple	0.995	5	0.5	0.02
cube-double, cube-quadruple, puzzle-3×3, puzzle-4×4, scene	0.99	5	0.5	0.02

I.4 Evaluation and aggregation

Every run is evaluated every 50k environment steps on 50 episodes, giving eleven points per curve including step 0, which evaluates the starting checkpoint. The evaluation seed is fixed and decoupled from the training seed, so every method and training seed sees the same 50 initial states; episode i uses the seed $(12345 \times 100003 + i) \bmod (2^{31} - 1)$. Pinning the initial state of an OGBench episode requires seeding three independent generators: the global NumPy generator (initial-position noise in the maze environments and the task draw), the environment's own generator (position and velocity noise at reset), and the action-space generator (stabilizing random actions after reset); the evaluation wrapper seeds all three. Training seeds are 10001, 20002, 30003, 40004, and 50005, and one seed initializes the Python, NumPy, and JAX generators.

A task curve is the mean of its five seed curves, a domain curve the mean of its five task curves, and the aggregate curve the mean of the ten domain curves; final scores are the values at 500k steps. The per-domain panels of Figure 3 show 95% t-intervals over the 25 seeds of a domain, its aggregate panel propagates the domain standard errors, and the ablation panels show ± 1 standard error across the 50 tasks. The efficiency axis of Figure 1 is the number of environment steps needed to first reach 50% success, and its error bars come from 2,000 bootstrap resamples of the 50 tasks. Table 4 lists the final success of every method in each domain.

Table 4: Final success (%) per domain after 500k steps, averaged over five tasks with five seeds each; the last column is the aggregate of Figure 3. Abbreviations: am, antmaze; hm, humanoidmaze; cube-2/3/4, cube-double/triple/quadruple; puz-3/4, puzzle-3×3/4×4.

Method	am-giant	am-large	hm-med.	hm-large	cube-2	cube-3	cube-4	puz-3	puz-4	scene	Agg.
SFEdit	95.9	94.7	85.3	74.6	100.0	67.4	79.8	100.0	80.0	98.5	87.6
SFEdit(QAM)	95.4	97.2	86.2	61.0	99.8	70.6	79.8	100.0	64.0	98.4	85.2
iQAM	55.6	97.9	90.7	62.0	99.9	74.2	71.5	100.0	64.0	99.0	81.5
QAM	53.9	97.4	87.1	74.1	99.8	76.1	59.8	100.0	59.9	98.6	80.7
TRQAM	71.0	98.5	86.6	45.8	100.0	76.9	77.0	100.0	42.2	99.1	79.7
EXPO	87.0	81.4	74.2	50.7	99.0	52.6	78.9	100.0	39.4	99.2	76.2
FQL	82.3	24.2	77.3	60.5	99.7	48.5	79.0	100.0	56.0	99.2	72.7
RLPD	85.9	97.2	61.7	0.0	72.1	4.1	75.0	99.9	100.0	89.9	68.6
DSRL	21.7	82.0	76.3	34.8	98.3	2.2	67.1	100.0	40.0	98.7	62.1
IFQL	15.8	63.3	49.2	7.8	72.3	5.3	12.9	100.0	44.9	98.3	47.0
BAM	0.6	68.4	10.6	35.3	96.9	29.8	10.7	85.2	16.1	81.9	43.5
QSM	65.4	18.7	83.3	0.0	0.0	0.1	0.0	0.0	0.0	1.4	16.9

J Ablation arms

All ablation curves aggregate the same 50 tasks with five seeds each. The κ sweep varies the movement scale of SFEdit, whereas the transport arms are matched to SFEdit(QAM), which shares its discretization with QAM, so that comparisons with QAM isolate the target and the transport.

Flow steps. The four arms of the first panel of Figure 4 are rerun with $K = 4$ and every other hyperparameter unchanged. Their final aggregates at ten and four steps are 87.6 and 58.6 for SFEdit, 85.2 and 24.0 for SFEdit(QAM), 81.5 and 55.9 for iQAM, and 80.7 and 43.1 for QAM. The envelope signal is thus more sensitive to the original recursion than the raw critic gradient, and iQAM, which

attenuates the coarse early intervals most (Appendix E), recovers 34.6 of the points that SFEdit(QAM) loses; we have not isolated which property of the signal causes this sensitivity.

Movement scale. The multipliers $\{0.25, 1, 2, 4\}$ are relative to each domain's default, so the absolute values are 0.05, 0.2, 0.4, and 0.8 in antmaze-giant and 0.005, 0.02, 0.04, and 0.08 elsewhere. The final aggregates are 86.5, 87.6, 80.7, and 71.5, against 80.7 for QAM. As $\kappa \to 0$, the transport tends to the identity and the target of the $M = 4$ solver to $\frac{15}{16}\beta\nabla Q$ (see below), so the family tends to iQAM with β scaled by $15/16$; iQAM at $\beta = 3$ reaches 81.5, and the QAM curve in Figure 4 is a reference only. Antmaze-giant is the one domain that prefers a larger scale than the others, which is why its default is ten times larger.

Transport with another learner. The QAM+transport arm attaches the solver of Appendix G.3 to QAM at the domain's default κ, $\rho_{\text{prox}} = 0.5$, $M = 4$, $\beta = 3$, and $K = 10$; it differs from SFEdit(QAM) only in the terminal adjoint signal. The RLPD+transport arm is not matched: it uses the unrelaxed iteration ($\rho_{\text{prox}} = 1$), and its κ was chosen per domain by final score. The final aggregates are 82.3 for QAM+transport against 80.7 for QAM, and 69.4 for RLPD+transport against 68.6 for RLPD. Per domain, the transport changes QAM much more than RLPD in both directions (mean absolute change 12.8 against 2.0 points, or 10.6 against 2.7 after removing the three domains where RLPD is at or near the floor or the ceiling), so this panel supports a sensitivity claim rather than a net-gain claim for RLPD.

Transport ablation. The no-transport arm keeps the envelope target and every solver setting but disables the transport at both execution and backup; the random arm replaces the direction of each displacement by a random direction with the magnitude of the actual proximal displacement. Both arms modify SFEdit(QAM), so their matched reference is SFEdit(QAM) (85.2), not SFEdit (87.6), which Figure 4 shows for context. The final aggregates are 82.8 for the no-transport arm and 0.2 for the random arm (0 in seven domains). Removing the transport costs 57.3 points in antmaze-giant (38.2 against 95.4, and 53.9 for QAM) and changes each of the other nine domains by a few points; averaged over those nine domains, success rises from 84.1 to 87.8, against 83.6 for QAM (Table 4). The transport therefore pays off where corrections are large and not, on average, where they are small. Two effects may lower its value there. Bootstrapping at edited actions evaluates the target critic near its local maxima, a source of maximization bias that the ensemble pessimism of the backup only partly offsets. With $M = 4$ and $\rho_{\text{prox}} = 1/2$, the approximate envelope gradient $(\widehat{T}_\kappa(a) - a)/\kappa$ also tends to $(1 - 2^{-4})\beta\nabla Q(a) = \frac{15}{16}\beta\nabla Q(a)$ as $\kappa \to 0$ when clipping is inactive, so the arms with the envelope target act at a slightly lower effective inverse temperature than QAM. Single-domain differences of about ten points arise even between QAM and iQAM, which differ in one scalar factor, so we read the nine-domain differences with caution, whereas the antmaze-giant effect far exceeds this spread. The random arm shows that perturbations of the proximal magnitude in arbitrary directions are destructive, so the transport acts through its direction; whether editing beats not editing is the question the no-transport arm answers. We did not run variants that apply the transport only at execution or only at backup.

K Real-robot setup and protocol

Hardware. A Franka Research 3 arm under 1 kHz Cartesian impedance control, a Robotiq 2F-85 gripper with binary open and close commands, a wrist-mounted Intel RealSense D405, and an external D435. Actions are absolute end-effector pose targets (position and Euler angles) plus a binary gripper command at a nominal 10 Hz. All computation runs on one machine with a single RTX 5060 Ti GPU, an Intel Core Ultra 5 245K CPU, and 32 GB of memory; the frozen $\pi_{0.5}$ policy server and the online learners share this GPU.

Steps and episodes. One step is one executed control step. Every policy query yields a ten-step chunk, of which the first eight steps are executed before the next query, identically for all methods. Training and evaluation episodes end after 200 steps for Screw Sorting and 250 for Plate Stacking; the episode-length comparison in Section 5.3 lifts this limit.

Teacher and demonstrations. The teacher is $\pi_{0.5}$ (Black et al., 2025a) fine-tuned for 10,000 gradient updates on 100 teleoperated demonstrations, 60 for Screw Sorting and 40 for Plate Stacking, and frozen afterwards. A green plate, whose color never appears in the demonstrations, is present during evaluation to test color matching beyond the demonstrated colors.

Student policy. The student is a velocity MLP with 8.63M parameters, about $1/50$ of the 427.9M-parameter action expert of $\pi_{0.5}$. Its conditioning c_s is the teacher's frozen pooled prefix representation (4,096 dimensions), passed through layer normalization and a dense projection to 1,024 dimensions and concatenated with 7 proprioceptive values (position, Euler angles, and gripper command). The field maps c_s, the noisy chunk x_t, and a Fourier embedding of t through four layers of width 1,024 with layer normalization to the 70-dimensional normalized chunk (10×7), predicts the action itself rather than a residual, and samples with four Euler steps from noise composed half of independent standard normal entries and half of a truncated normal draw broadcast along the chunk. It is distilled by paired flow matching: at each demonstration observation the teacher is sampled with ten Euler steps from 16 fixed noise draws, and the student regresses the velocity along the straight path from each noise draw to its teacher action, with about 25% of episodes held out. Training uses 60k steps, batch 256, Adam at 3×10^{-4}, gradient clipping 1.0, an exponential moving average of the weights with rate 0.999, and a loss weight of 0.25 on the gripper coordinate; the endpoint mean squared error in normalized coordinates is 0.003 on the training pairs and 0.014 to 0.018 on the held-out pairs, and distillation takes about three hours per task on the robot's computer.

Online learning, all methods. Each chunk receives reward -1, and the final chunk of a successful episode receives 0 and terminates; a human operator judges success. Ten warm-up episodes are collected without updates, followed by a first training block and then one training block every two episodes, with a number of updates equal to the update-to-data ratio times the number of new transitions. The replay buffer holds online data only (capacity 100k, uniform sampling), so the flow-matching term of SFEdit also regresses the student toward the executed, transported actions. The critic has two heads (MLP $1{,}024 \times 3$ with layer normalization), the target takes the minimum over heads, the Polyak rate is 0.005, and the critic batch size is 256. Safety checks reject a chunk that would leave the workspace ($|x|, |y| \leq 0.6$ m, with a task-specific lower bound on z that prevents unsafe gripper closure) or whose per-step displacement exceeds the limit in Table 5.

Evaluation. Each method receives a 45-minute online training budget per run and three independent runs per task, and each final policy is evaluated on 30 episodes with randomized object positions. The teacher and the distilled student are evaluated before adaptation. SFEdit samples the student and applies the transport, DSRL executes $\tanh$ of its mean latent, and RLT executes its mean action.

The robot-side movement scale and temperature ($\kappa = 0.05$, $\beta = 0.3$) differ from the simulation values ($\kappa = 0.02$, $\beta = 3$) because the action coordinates and the critic scale differ; both were fixed after preliminary tests and before the reported runs, and QAM uses the same β.

Table 5: Real-robot online learning settings per method. All methods share the reward, discount, replay, critic architecture, warm-up, and safety checks above; QAM uses the SFEdit column without the envelope target, transport, and iQAM.

	SFEdit	DSRL	RLT
Policy	student flow with proximal transport	SAC over the teacher's initial noise	actor conditioned on the teacher chunk
State	c_s	pose and gripper (7) and prefix feature (2,048)	as DSRL
Actor	student, 8.63M, Adam 3×10^{-5}, clip 1.0, batch 64	MLP $1{,}024 \times 3$, 4.27M, 10^{-4}	MLP $1{,}024 \times 3$, 4.35M, 10^{-4}, exploration std 0.02
Critic learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Method-specific	iQAM; $\mathcal{L}_{\mathrm{FM}} + 0.3\mathcal{L}_{\mathrm{AM}}$; $\kappa = 0.05$, $\beta = 0.3$, $M = 4$; reference fixed to the distilled student	target entropy 0, temperature learning rate 3×10^{-4}	BC weight 25, reference dropout 0.5, anchor radius 0.5, two critic updates per actor update
Update-to-data ratio	10 (screws), 5 (plates)	10	5
Warm-up steps	5,000	5,000	5,000
Warm-up behavior	unedited student samples	teacher with standard normal noise	teacher
Movement limit	0.06 m, 0.20 rad	0.10 m, 0.45 rad	0.10 m, 0.45 rad