BRANCHPO: DEMONSTRATION-ANCHORED BRANCHING PREFERENCE OPTIMIZATION FOR VLA POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained vision–language–action policies can remain unreliable on multi-stage deployment tasks, while collecting expert corrections at learner-visited states is expensive and from-start autonomous rollouts can undersample later phases. We present BRANCHPO, a post-training framework that reuses one successful demonstration as a phase-indexed interaction scaffold. Semantic roots provide access to distinct task phases; autonomous sibling continuations are collected from each root and compared with cached expert segments. For flow-matching policies, BranchPO optimizes a root-balanced, reference-relative energy preference objective with explicit expert attraction, requiring neither a learned critic nor additional expert action labels. An idealized analysis characterizes root-local outcome contrast, heterogeneous phase corrections, and the comparison weights retained by the implemented loss. Across RoboCasa and real-robot manipulation tasks, BranchPO improves the SFT initialization on every evaluated task and generally outperforms AWR and trajectory-level DPO. Scaling from-start whole-trajectory preferences does not recover the benefit of rooted continuations, while a positive-only adaptation control explains only a small part of the gain. These results show that demonstration structure can guide both autonomous data collection and local credit assignment for effective VLA post-training.

1 INTRODUCTION

Vision–language–action (VLA) policies acquire broad manipulation capabilities from large-scale demonstrations, yet can remain unreliable on a particular multi-stage deployment task (Black et al., 2025; Physical Intelligence et al., 2025). Supervised adaptation on a few task demonstrations continues to train on expert-visited states. Autonomous post-training can instead expose the policy to its own execution distribution (Amin et al., 2025; Chen et al., 2025), but its value depends on which states are reached and how credit is assigned to the resulting behavior.

From-start rollouts create a reachability bottleneck: experience from a later phase appears only when all preceding phases succeed. Collecting more complete episodes can therefore repeat early failures without improving coverage of the decisions that follow. A single whole-trajectory preference also couples actions from several phases into one comparison. Targeted expert intervention can address these problems, as in DAgger (Ross et al., 2011), but requires new expert actions at learner-visited states. We ask whether one successful demonstration can instead organize autonomous experience and provide phase-local supervision without further expert action labels.

We present BRANCHPO, a demonstration-anchored framework for post-training flow-based VLAs. The demonstration serves two roles: semantic roots provide access to distinct task phases, and the corresponding cached segments provide local expert references. From each root, the policy executes multiple autonomous sibling continuations over a comparison window. The siblings and their expert segment form a *root block* (Figure 1). Rooted collection changes the visited-state distribution itself; post-hoc segmentation can partition a from-start rollout, but cannot recover a phase that the policy never reached.

For flow-matching policies, BranchPO compares reference-relative energies between each sibling and the expert segment from the same root. It applies the preference nonlinearity before averaging,

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

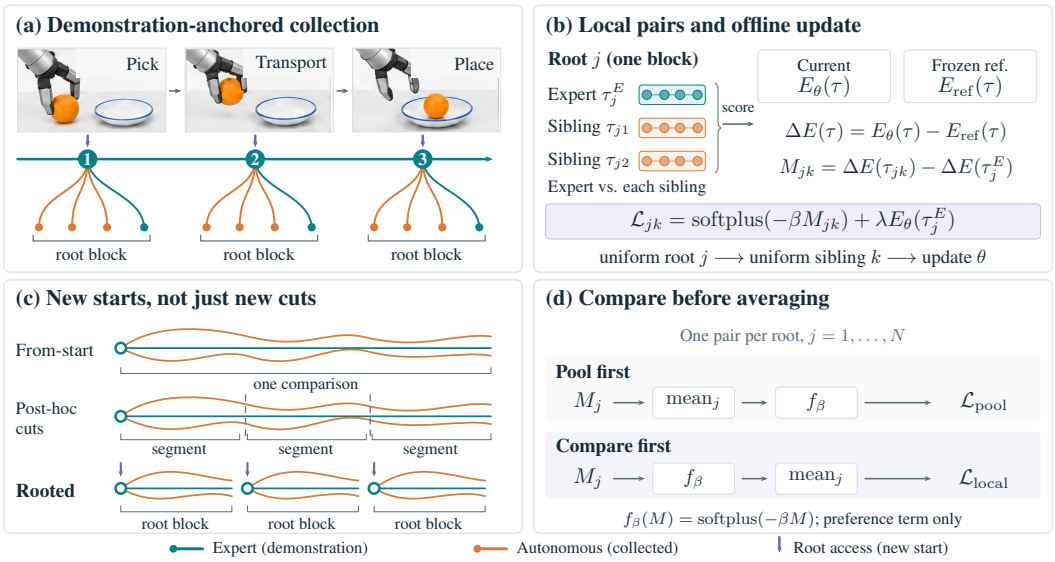


Figure 1: **BranchPO: rooted collection and local preference learning.** **(a)** One demonstration supplies phase roots and expert segments; repeated continuations form root blocks. **(b)** Each sibling is compared with its root's expert segment using reference-relative flow energies ($E_{\text{ref}} = E_{\theta_0}$), followed by a root-balanced offline update. **(c)** Root access changes where experience starts; post-hoc cuts only partition existing rollouts. **(d)** Root-wise comparison applies the preference loss before aggregation (one pair per root shown; expert attraction omitted). Teal/orange encode expert/autonomous provenance, not outcomes. Robot scenes are illustrative.

balances roots, and retains an explicit expert-attraction term. This construction requires neither a learned critic nor additional expert action labels. Its contribution is the coupling of phase-indexed collection with local comparison, rather than a new normalized likelihood objective. The coupling yields a concrete empirical prediction: increasing from-start, whole-trajectory preference data alone should not recover the benefit of rooted access and comparison.

Our contributions are:

- **Demonstration-anchored autonomous collection.** We introduce root blocks, which reuse one successful demonstration to access multiple task phases and pair cached expert segments with autonomous local alternatives.
- **Root-local flow preference learning.** We develop a root-balanced, reference-relative flow-energy objective and analyze ideal outcome contrast, heterogeneous phase corrections, and the margin-dependent comparison weights of the implemented loss.
- **Evaluation in simulation and on real robots.** We compare BranchPO with SFT-only, AWR, Trajectory-DPO, and positive-only adaptation. BranchPO improves the SFT initialization across the evaluated tasks; scaling whole-trajectory preferences does not reproduce the rooted gain, and real-robot evaluation shows improved multi-stage task progression.

2 Problem Setup and Preliminaries

2.1 Policy and Task

We use a finite-horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, H)$ with decision index h. The VLA samples an action plan $a_h \sim \pi_\theta(\cdot \mid s_h)$, and the controller executes its prefix under a fixed replanning rule. This execution rule, including truncation at comparison-window boundaries, is part of P. Here s_h is an information state containing the visual observation, instruction, and any history required for the Markov abstraction. We write Q_h^π, V_h^π and $A_h^\pi = Q_h^\pi - V_h^\pi$ for the standard finite-horizon value and advantage functions. The goal is task-success improvement under a fixed robot, task and scene with

deployment-time pose and execution variation. The pretrained flow VLA $\pi_{\text{ref}} = \pi_{\theta_0}$ supplies both initialization and frozen reference; π_b is the collection policy.

Let τ^E be a successful demonstration recorded at controller resolution. Action-plan targets are reconstructed separately within each rooted window, rather than on one global query grid. An access procedure establishes selected phases along this trajectory. The demonstration supplies phase anchors and preferred segments, not expert labels at every nearby state. Indices j, k and ℓ denote roots, siblings and segment positions; E marks expert data and undecorated segments are autonomous.

2.2 Flow-matching segment scores

We reserve h for policy-decision time and use $t \in [0, 1]$ only for flow time. Under the data-to-noise interpolation used by our implementation, $x_t = (1 - t)a + t\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, so the conditional target field is $u_t(a, \epsilon) = \partial_t x_t = \epsilon - a$. The flow head is denoted $v_\theta(x_t, t \mid s)$. To score a logged state–action pair, we use the flow-matching energy

$$e_\theta(s, a) = \mathbb{E}_{t\sim p_{\text{FM}},\, \epsilon\sim\mathcal{N}(0,I)} \left[\text{MSE}_{\text{exec}}(v_\theta(x_t, t \mid s), u_t(a, \epsilon))\right]. \tag{1}$$

Here p_{FM} is inherited from pretraining. The logged execution mask in MSE_{exec} restricts the mean to physical action dimensions and actually executed positions; corruption still applies to the full native plan. For a segment of L local queries $\tau = ((s_\ell, a_\ell))_{\ell=0}^{L-1}$ with deployment weights $w_\ell \geq 0$ and $\sum_{\ell=0}^{L-1} w_\ell = 1$, let

$$E_\theta(\tau) = \sum_{\ell=0}^{L-1} w_\ell e_\theta(s_\ell, a_\ell), \qquad \Delta E_\theta(\tau) = E_\theta(\tau) - E_{\theta_0}(\tau). \tag{2}$$

This flow-matching residual is a compatibility score, not a normalized action log-likelihood (Lipman et al., 2023; Black et al., 2025). Its use in a preference loss gives a DPO-like surrogate (Rafailov et al., 2023; Wu et al., 2026).

3 BranchPO

BranchPO (*Branching Preference Optimization*) couples two decisions: where autonomous experience starts and which segment receives a preference comparison (Figure 1). We describe the frozen-data, segment-level method used in the experiments.

3.1 Demonstration-anchored branching

Select N task-phase roots along the demonstration and a window of W_j control steps at each root. The recorded expert behavior supplies τ_j^E. Starting independently from the same root, π_b executes an equally long window to generate a sibling. Both segments are represented by local action-plan queries under the same replanning rule, with the final executed prefix truncated at the window boundary. Scripted expert-event rules select roots in the representative experiments; windows may overlap (Appendix B.2). Repeated collection yields

$$\mathcal{B}_j = \{\tau_j^E, \tau_{j1}, \ldots, \tau_{jK}\}, \tag{3}$$

where τ_{jk} is the kth autonomous continuation. The two segments in a comparison share a root condition and window length, not their subsequent states or task progress. In particular, the expert action at one offset is not treated as a dense correction for the sibling's state at that offset.

Roots control phase coverage; K controls alternatives per phase. The handoff distribution ν_j is induced by root access, not the policy's natural state occupancy. A single demonstration anchors these neighborhoods without representing the deployment distribution.

Rooting is not post-hoc segmentation. Post-hoc cuts cannot recover unvisited phases. BranchPO independently establishes each root, so collecting a later-phase sibling does not require earlier success on that trial (Figure 1c). An autonomous endpoint is not the next root.

Algorithm 1 BranchPO: frozen-data, segment-level post-training

Require: reference $\pi_{\text{ref}} = \pi_{\theta_0}$, demonstration τ^E, roots and windows $\{W_j\}$, siblings per root K, root access

$\theta \leftarrow \theta_0$; freeze reference and collection policy $\pi_b = \pi_{\text{ref}}$
for $j = 1, \ldots, N$ **do**
  construct root-local expert reference τ_j^E
  for $k = 1, \ldots, K$ **do**
    establish a valid root; prepare the paired expert context; record access cost
    execute π_b for W_j control steps to obtain τ_{jk}
  end for
  form $\mathcal{B}_j$ with the expert preferred to each autonomous sibling
end for
for each optimization step **do**
  sample a root j uniformly, then a sibling k uniformly within $\mathcal{B}_j$
  draw shared flow time t and noise ϵ for compatible score evaluations
  estimate M_{jk} and $E_\theta(\tau_j^E)$; update θ using equation 5
end for
return adapted policy π_θ

3.2 Root-balanced flow preference

Each expert–sibling pair has reference-relative margin

$$M_{jk}(\theta) = \Delta E_\theta(\tau_{jk}) - \Delta E_\theta(\tau_j^E). \tag{4}$$

A positive margin favors the expert relative to the frozen reference. The pair loss is

$$\mathcal{L}_{jk}(\theta) = \text{softplus}(-\beta M_{jk}(\theta)) + \lambda E_\theta(\tau_j^E), \qquad \beta > 0, \quad \lambda = 0.1. \tag{5}$$

The margin can improve while both energies increase; the second term therefore supplies direct expert attraction. It is a fit penalty, not a policy KL constraint.

Uniform root sampling gives

$$\mathcal{L}_{\text{BPO}}(\theta) = \frac{1}{NK} \sum_{j=1}^{N} \sum_{k=1}^{K} \mathcal{L}_{jk}(\theta). \tag{6}$$

For unequal counts, each inner average uses its own K_j, keeping phase weight independent of sibling count. Let $\mathcal{P}_{jk} = \text{softplus}(-\beta M_{jk})$ and denote its average by $\mathcal{L}_{\text{pref}}$. Crucially, comparisons enter the nonlinearity *before* aggregation, preserving separate pair weights (Section 4.2). Within-root averaging keeps expert attraction from scaling with repeated use of a positive.

At initialization, $M_{jk} = 0$: descent attracts expert energies and repels sibling energies (Appendix B). This positive–negative form is related to DiffusionNFT (Zheng et al., 2026), not its forward-regression objective.

3.3 Optimization and scope of credit

Algorithm 1 uses $\pi_b = \pi_{\text{ref}}$ and keeps the collected dataset and reference fixed. Each update samples a root-balanced pair and shared flow-time/noise draws, then evaluates equation 5 on the resulting sampled energies.

One label acts across a root window, not independently at each transition: BranchPO localizes comparison scope without providing dense expert corrections. In simulation, expert observations are rendered in the sibling-matched context while the demonstrated actions are reused; τ_j^E suppresses this dependence. Appendix B specifies query reconstruction, execution masks, hyperparameters, and representative root protocols.

4 Local Analysis of Rooted Preference Learning

We connect root-local outcome information to policy improvement, then examine phase-specific corrections and the weighting induced by pairwise comparison.

4.1 ROOT-LOCAL POSTERIOR CONTRAST

Use undiscounted return with zero intermediate rewards and terminal reward $Y \in \{0, 1\}$, so $G_h = \sum_{u=h}^{H-1} r_u = Y$ and $\mathbb{E}[G_h \mid \cdot] = \mathbb{E}[Y \mid \cdot] = \Pr(Y = 1 \mid \cdot)$. This probability interpretation follows from the binary reward definition, not from normalizing an arbitrary return. At a fixed rooted decision (h, s), the standard finite-horizon values are

$$\begin{aligned} Q(a) &:= Q_h^{\pi_{\text{ref}}}(s, a) = \mathbb{E}_{\pi_{\text{ref}}}[G_h \mid s_h = s, a_h = a] = \Pr_{\pi_{\text{ref}}}(Y = 1 \mid s_h = s, a_h = a), \\ V &:= V_h^{\pi_{\text{ref}}}(s) = \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)}[Q(a)] = \Pr_{\pi_{\text{ref}}}(Y = 1 \mid s_h = s). \end{aligned} \tag{7}$$

Let π^+ and π^- be the reference action distributions conditioned on success and failure at this root. For analysis, use normalized-policy scores $\psi(a) = \nabla_\theta \log \pi_\theta(a \mid s)|_{\theta_0}$ and write $\mathcal{J}(\theta) = \mathbb{E}_{\pi_\theta}[Q(a)]$, $g = g_h(s) = \nabla \mathcal{J}(\theta_0)$. Only the root action changes; subsequent decisions follow π_{ref}. Appendix A.1 gives the Bayes forms, and Appendix A.3 relates them to the flow-field mixture underlying NFT (Zheng et al., 2026).

Proposition 1 (One-positive posterior contrast)**.** *Assume valid differentiation under the policy integral, $\mathbb{E}_{\pi_{\text{ref}}}[\psi] = 0$, and $\mathbb{E}_{\pi^\pm}\|\psi\|_2^2 < \infty$. Let $\kappa = V(1 - V)$, with $0 < V < 1$. For independent $a^+ \sim \pi^+$ and $a_1^-, \ldots, a_K^- \overset{\text{iid}}{\sim} \pi^-$, the empirical contrast $\widehat{g}_K = \kappa(\psi(a^+) - K^{-1} \sum_k \psi(a_k^-))$ is unbiased, with*

$$g = \kappa\big(\mathbb{E}_{\pi^+}[\psi] - \mathbb{E}_{\pi^-}[\psi]\big), \tag{8}$$

$$\mathrm{Cov}[\widehat{g}_K] = \kappa^2(\Sigma^+ + \Sigma^-/K), \tag{9}$$

where $\Sigma^\pm = \mathrm{Cov}_{\pi^\pm}[\psi]$. If $\mathcal{J}$ is L-smooth and $g \neq 0$, a sufficiently small step $\theta_0 + \eta \widehat{g}_K$ improves $\mathcal{J}$ in expectation. Appendix A.1 gives the bound and stepsize condition.

The expectation covers both positive and negative draws. More siblings reduce negative-side variance but leave the positive-side floor Σ^+. Moreover, $\mathcal{J}(\theta) - \mathcal{J}(\theta_0) = \mathbb{E}_{\pi_\theta}[A_h^{\pi_{\text{ref}}}(s, a)]$: root-local improvement is positive expected reference advantage for the one-step deviation.

4.2 ROOT-CONDITIONED CORRECTIONS AND LOCAL PREFERENCE WEIGHTING

Value of phase-specific corrections. Let $g_j = \mathbb{E}_{s \sim \nu_j}[g_{h_j}(s)]$ and $\bar{g} = N^{-1} \sum_j g_j$. Consider an ideal root-gated family with root-specific copies $\Delta\theta_j$ of reference-policy coordinates. For a common positive-definite metric F, such as a damped Fisher matrix, impose $N^{-1} \sum_j \|\Delta\theta_j\|_F^2 \leq \delta^2$. The corresponding dual norm is $\|g\|_{F^{-1}}^2 = g^\top F^{-1} g$; Appendix A.2 states the optimization problems explicitly.

Proposition 2 (Root-conditioned improvement potential)**.** *For $\delta > 0$, let Γ_{root} be the largest root-averaged first-order gain in this family, and Γ_{pool} the largest gain under the tied restriction $\Delta\theta_1 = \cdots = \Delta\theta_N$. Then*

$$\Gamma_{\text{root}} = \delta \sqrt{\frac{1}{N} \sum_{j=1}^{N} \|g_j\|_{F^{-1}}^2}, \tag{10}$$

$$\Gamma_{\text{pool}} = \delta \|\bar{g}\|_{F^{-1}}, \tag{11}$$

and $\Gamma_{\text{root}} \geq \Gamma_{\text{pool}}$, with squared gap

$$\frac{\Gamma_{\text{root}}^2 - \Gamma_{\text{pool}}^2}{\delta^2} = \frac{1}{N} \sum_{j=1}^{N} \|g_j - \bar{g}\|_{F^{-1}}^2 \geq 0. \tag{12}$$

Equality holds if and only if all root-local gradients are identical.

The cost of tying corrections grows with root-gradient dispersion: $\bar{g} = 0$ can leave no tied first-order gain even when individual roots admit improvement.

Root-wise weighting in the implemented loss. BranchPO applies the preference nonlinearity to each pair before averaging. With $f_\beta(M) = \mathrm{softplus}(-\beta M)$, the negative margin derivative for pair (j, k) is

$$-\frac{\partial \mathcal{L}_{\mathrm{pref}}}{\partial M_{jk}} = \frac{\beta}{NK}\sigma(-\beta M_{jk}). \tag{13}$$

Thus any non-positive margin retains weight at least $\beta/(2NK)$, irrespective of satisfied comparisons elsewhere. A mean-pooled comparator instead assigns one common coefficient after margins have been aggregated, allowing positive margins to hide a violated root. The weights coincide at initialization and diverge as comparisons become heterogeneous. Appendix A.4 gives the pooled objective, Jensen bounds, and the corresponding geometry.

5 Experiments

We first compare task success across simulation and real robots, then examine whether the gains can be reproduced by scaling whole-trajectory preferences or repeating positive-only adaptation. Finally, qualitative real-robot executions illustrate where task progression changes.

Our simulation study covers five kitchen manipulation tasks from RoboCasa (Nasiriany et al., 2026): *OvenBroilFish*, *PastryDisplay*, *FoodCleanup*, *RemoveBroiledFish*, and *LoadDishwasher*. These tasks span food placement, organization, retrieval, and dishwasher loading, requiring policies to complete multi-stage manipulation sequences. Real-robot experiments cover three tasks: placing a banana on the designated red plate, retrieving a tissue after closing the box lid, and storing blocks in a drawer before closing it. Across both settings, we evaluate complete task execution rather than isolated manipulation primitives.

Simulation starts from the official $\pi_{0.5}$ RoboCasa checkpoint (Physical Intelligence et al., 2025), and BranchPO uses one successful demonstration per task. The main comparison includes SFT-only, AWR (Peng et al., 2019), Trajectory-DPO (Rafailov et al., 2023), and BranchPO, with 50 evaluation episodes per method and task in simulation and 20 trials on each real-robot task. Evaluation varies initial robot/object poses within fixed simulation scenes and object positions and orientations on the real robot. Training and evaluation settings are detailed in Appendix C.

5.1 Task performance across simulation and real robots

BranchPO improves pretrained policies through demonstration-anchored autonomous rollouts, without additional expert action labels. A single demonstration supplies reusable phase-local comparison references, while autonomous execution supplies alternative behaviors without requiring expert corrections at newly visited states (Algorithm 1). The success–failure contrast in Proposition 1 provides an idealized local improvement rationale under matched outcome-conditioned sampling. With the practical source-labeled flow objective in equation 5, BranchPO consistently improves task success over SFT-only across simulation and real robots (Table 1). These results show that a cached demonstration can serve as a reusable reference for effective post-training on autonomous experience, without expanding expert action supervision.

Table 1: **Task success across simulation and real-world manipulation.** Each reported simulation checkpoint is evaluated for 50 episodes per task. Higher is better. Green (bold) and blue backgrounds mark the best and second-best results per task, respectively.

	Simulation (50 episodes/task)					*Real robot (20 trials/task)*		
Method	Oven broil	Pastry display	Food cleanup	Remove fish	Load dishwasher	Banana placement	Tissue retrieval	Drawer storage
SFT-only	2%	38%	28%	6%	24%	15%	5%	0%
AWR	22%	46%	18%	**34%**	22%	45%	25%	5%
Trajectory-DPO	6%	34%	10%	12%	12%	5%	0%	0%
BranchPO	**72%**	**96%**	**64%**	28%	**42%**	**95%**	**90%**	**85%**

5.2 ROOTED VERSUS FROM-START PREFERENCES

BranchPO retains phase-local optimization pressure that whole-trajectory aggregation can obscure. We compare BranchPO with a trajectory-level DPO baseline (Rafailov et al., 2023) (Trajectory-DPO), implemented with the same flow-energy surrogate and expert-attraction term as BranchPO. It favors the expert demonstration over autonomous trajectories through one preference margin per full trajectory pair, whereas BranchPO makes separate comparisons within task phases. This comparison asks whether rooted, phase-local preference learning offers an advantage beyond simply increasing the amount of autonomous experience. Table 2 evaluates this question on PastryDisplay.

Table 2: **PastryDisplay: rooted branching versus scaled whole-trajectory preferences.** Over the same 50 evaluation cases, gained successes count SFT-only failures converted to successes, while lost successes count SFT-only successes converted to failures. Blue shading tracks increasing data volume; warm cells emphasize trajectory-level outcomes, and the green row highlights BranchPO.

	Collection budget		*Evaluation outcomes*		
Method	Autonomous data	Action frames	Success	Gained successes ↑	Lost successes ↓
SFT-only	–	–	19/50 (38%)	–	–
Trajectory-DPO	12 full traj.	4,008 (**0.98**×)	19/50 (38%)	11	11
Trajectory-DPO	64 full traj.	21,376 (**5.24**×)	17/50 (34%)	11	13
Trajectory-DPO	128 full traj.	42,752 (**10.48**×)	3/50 (6%)	0	16
Trajectory-DPO	256 full traj.	85,504 (**20.96**×)	0/50 (0%)	0	19
BranchPO	64 rooted seg.	4,080 (1.00×)	48/50 **(96%)**	**29**	**0**

BranchPO substantially outperforms Trajectory-DPO at both comparable and larger baseline data budgets (Table 2). The paired outcomes reveal that this gain reflects an expansion of successful execution: BranchPO solves previously failed cases while retaining all SFT-only successes on this evaluation set, whereas Trajectory-DPO's gains are offset or exceeded by losses of previously successful cases. These results support rooted comparisons as an effective way to turn autonomous experience into corrective supervision, beyond increasing the volume of whole-trajectory preferences.

Rooted comparisons retain phase-specific corrective pressure within the preference objective. This contrast is consistent with the rationale in Section 4.2. In its ideal root-gated model, equation 12 quantifies the first-order improvement potential lost by tying heterogeneous phase corrections. For the implemented loss, equation 13 shows that a violated root-local comparison retains its own margin-gradient weight, regardless of satisfied comparisons elsewhere. Trajectory-DPO also weights comparison pairs, but at the full-trajectory level; BranchPO applies this pressure separately within each root. Increasing rollout count does not change that comparison scope.

5.3 POSITIVE-ONLY ADAPTATION CONTROL

The observed refinement extends beyond repeated fitting of the demonstration. The objective in equation 5 couples direct attraction to the expert segment with the sibling-relative margin M_{jk}. At the reference policy, the latter produces both expert-attraction and sibling-repulsion components in its exact energy-ranking descent direction (equation 40). One-demo SFT receives the same anchor trajectory but does not form an autonomous sibling contrast. It is therefore an empirical proxy for the anchor-only route, rather than a term-by-term ablation of the $\lambda E_\theta(\tau_j^E)$ term.

Table 3 summarizes a separate OvenBroilFish comparison using a different BranchPO run from Table 1. One-demo SFT improves on SFT-only by 8 percentage points, while BranchPO exceeds the positive-only control by 52 percentage points. Under this recipe, repeated imitation of the same demonstration therefore recovers only a small part of the observed BranchPO gain.

Table 3: **Positive-only adaptation control on OvenBroilFish.** 50 episodes per policy.

Method	SFT-only	One-demo SFT	BranchPO
Task success	1/50 (2%)	5/50 (10%)	31/50 **(62%)**

5.4 Real-robot evaluation: closed-loop task progression

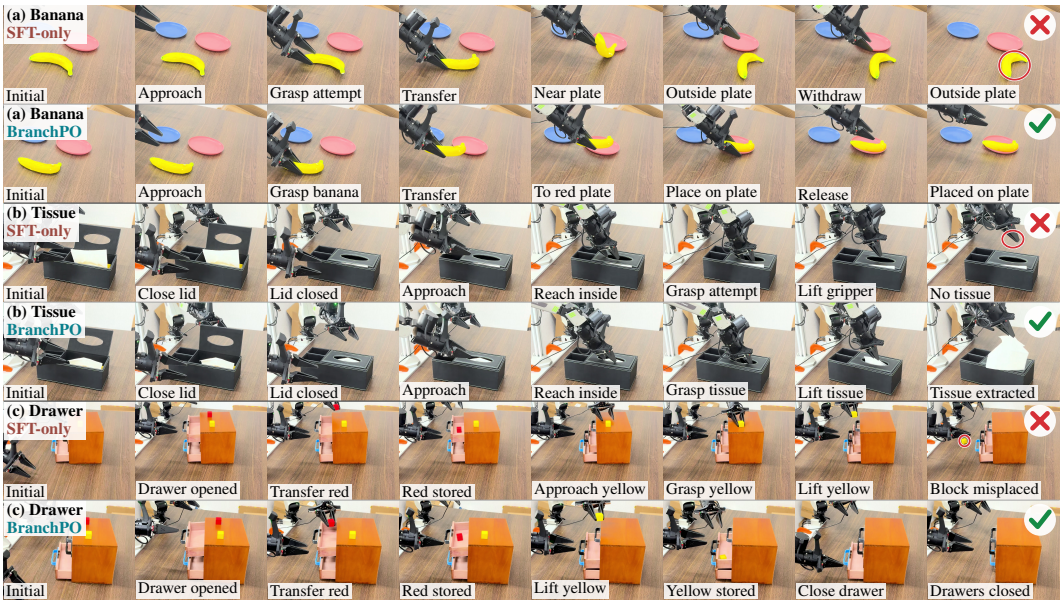


Figure 2: **Selected SFT-only failures and BranchPO successes.** These qualitative examples illustrate refinement relative to the SFT initialization, with SFT-only on top and BranchPO on the bottom for each task: **(a)** banana placement on the red plate, **(b)** tissue retrieval, and **(c)** drawer storage. Read eight keyframes left to right. Green checks/red crosses mark trial success/failure; red outlines highlight the empty grasp or misplaced objects. Each row shows an independently recorded execution. The quantitative detailed comparison is reported in Table 1.

BranchPO sustains task progress beyond milestones that SFT-only already reaches. In the selected banana-placement example in Figure 2, SFT-only leaves the object outside the target plate; in tissue retrieval, it closes the lid but lifts the gripper without extracting a tissue; in drawer storage, it stores the red block but leaves the yellow block on the table. BranchPO completes the corresponding interactions. Together with the full-task results in Table 1, these examples indicate that the gain extends beyond reaching an early milestone to sustaining the later interactions needed for completion. They also show why the comparison unit in equation 3 is an executed continuation: segments sharing a starting phase can diverge in both visited states and achieved progress.

This pattern is consistent with phase-specific correction in Proposition 2. Rooted continuations expose the policy to states induced by its own actions, while the phase-matched expert segment supplies a local reference. The pairwise weighting in equation 13 then preserves pressure on an unsatisfied comparison even when another phase is already well ranked. BranchPO can therefore learn from execution deviations without requesting new expert actions at the deviated states.

6 Related Work

Interactive and autonomous robot post-training. Robot policies can be refined through reinforcement learning, intervention, or autonomous experience after imitation pretraining. RECAP combines a learned value function with advantage-conditioned VLA training (Amin et al., 2025); PA-RL uses

a critic to improve sampled actions and distills them into expressive policies (Mark et al., 2024). ConRFT combines consistency-policy learning, value estimation, and online human intervention (Chen et al., 2025), while HIL-SERL integrates demonstrations, corrections, and off-policy RL for physical manipulation (Luo et al., 2024). DSRL instead steers a frozen diffusion or flow policy through its input noise (Wagenmaker et al., 2025). These approaches span learned critics, additional interaction-time feedback, and test-time steering. BranchPO studies a complementary setting: a frozen autonomously collected dataset, no learned critic, and no new expert actions beyond the anchor demonstration.

Recent VLA post-training work further separates system, optimization, and data-generation choices. SOP couples a fleet of physical robots to a centralized learner for distributed online post-training (Pan et al., 2026). TEMPO updates the semantic projection and action expert on different RL timescales (Liu & Yang, 2026), while Cao et al. (2026) isolate how critic-derived advantages are constructed, calibrated, and weighted. PLD trains lightweight residual RL specialists to probe VLA failure regions, then distills curated recovery trajectories back into the generalist with SFT (Xiao et al., 2026). These methods target scalable online learning, optimization stability, or autonomous data generation. BranchPO instead asks how one cached successful demonstration can structure phase coverage and segment-level preference credit in a frozen offline update.

Demonstrations as state-access structure. Interactive imitation methods such as DAgger address compounding error by querying the expert at learner-visited states (Ross et al., 2011). Backplay instead uses demonstration states to construct a reverse initialization curriculum (Resnick et al., 2018). Self-Replay and MILES further show that a single demonstration can organize autonomous robot learning through reset, replay, or self-supervised expansion (Di Palo & Johns, 2022; Papagiannis & Johns, 2025). BranchPO uses a demonstration for two coupled purposes: its states provide phase access, and its aligned segments provide local preference references. Autonomous siblings may diverge after a shared root, so the method does not assume dense expert actions at their visited states. It also collects each phase independently; post-hoc segmentation of from-start episodes cannot create experience from an unreached phase.

Preference learning for diffusion and flow policies. DPO learns from relative preferences without fitting an explicit reward model (Rafailov et al., 2023), and Diffusion-DPO adapts this principle to diffusion models (Wallace et al., 2024). DiffusionNFT and FlowPRO bring related positive–negative signals to diffusion or flow policies (Zheng et al., 2026; Wu et al., 2026). FlowPRO is especially close in robotics: intervention and rollback produce correction pairs, and missing action counterparts are synthesized to obtain per-state flow preferences. BranchPO instead forms source-labeled comparisons between a cached expert segment and an autonomous continuation over a complete root window. It neither synthesizes expert actions for divergent sibling states nor claims transition-level correction. The novelty is therefore the coupled data-collection and credit-assignment recipe, rather than a fundamentally new preference loss; our Trajectory-DPO baseline tests the principal whole-trajectory alternative under the same flow-energy surrogate.

Flow-based post-training can also refer to constructing an action model rather than improving a deployed policy from interaction. LabVLA, for example, follows action-token pretraining with flow-matching post-training of a DiT action expert for scientific laboratory protocols (Ren et al., 2026). BranchPO begins from an already pretrained flow VLA and focuses on interaction-derived preference adaptation.

7 Conclusion

We present BranchPO, a VLA post-training method that uses one successful demonstration to structure autonomous interaction. Semantic roots localize data collection and credit, while cached segments provide fixed references for critic-free flow-energy preference updates. Experiments in simulation and on real robots show improvements over SFT, with rooted continuations proving more effective than larger from-start preference datasets. These results highlight the importance of where experience begins and how credit is assigned.

AI Assistance Disclosure

Large language models were used to assist with drafting, revising, and polishing the manuscript; retrieving and discovering relevant literature; and supporting research ideation as well as methodological discussion. All AI-generated suggestions and materials were critically reviewed and verified by the authors before inclusion. The authors take full responsibility for the correctness, originality, references, and scientific claims of the paper.

References

Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, et al. $\pi^*_{0.6}$: A VLA That Learns From Experience. *arXiv preprint arXiv:2511.14759*, 2025. URL https://arxiv.org/abs/2511.14759.

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 *Robotics: Science and Systems*, 2025. URL https://www.roboticsproceedings.org/rss21/p010.html.

Jiahang Cao, Hanye Zhao, Hang Lai, Shenyu Zhang, Xiaoshen Han, Xinghang Li, Futeng Liu, Wanli Peng, Heyun Wang, Yunhong Wang, Jason Li, Yong Yu, and Weinan Zhang. Dissecting advantage-guided post-training for vision-language-action policies. *arXiv preprint arXiv:2609.28161*, 2026. URL https://arxiv.org/abs/2609.28161.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. ConRFT: A reinforced fine-tuning method for VLA models via consistency policy. In *Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.019. URL https://www.roboticsproceedings.org/rss21/p019.html.

Norman Di Palo and Edward Johns. Learning multi-stage tasks with one demonstration via self-replay. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1180–1189. PMLR, 2022. URL https://proceedings.mlr.press/v164/palo22a.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Ziheng Liu and Quantao Yang. TEMPO: Semantic-action decoupled RL post-training for vision-language-action models. *arXiv preprint arXiv:2608.07314*, 2026. URL https://arxiv.org/abs/2608.07314.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024. URL https://arxiv.org/abs/2410.21845.

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.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots. *arXiv preprint arXiv:2603.04356*, 2026. URL https://arxiv.org/abs/2603.04356.

Mingjie Pan, Siyuan Feng, Qinglin Zhang, Xinchen Li, Jianheng Song, Chendi Qu, Yi Wang, Chuankang Li, Ziyu Xiong, Zhi Chen, Yi Liu, and Jianlan Luo. SOP: A scalable online post-training system for vision-language-action models. *arXiv preprint arXiv:2601.03044*, 2026. URL https://arxiv.org/abs/2601.03044.

Georgios Papagiannis and Edward Johns. MILES: Making imitation learning easy with self-supervision. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 810–829. PMLR, 2025. URL https://proceedings.mlr.press/v270/papagiannis25a.html.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. URL https://arxiv.org/abs/1910.00177.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Baochang Ren, Xinjie Liu, Xi Chen, Yanshuo Liu, Chenxi Li, Daqi Gao, Zeqin Su, Jintao Xing, Zirui Xue, Rui Li, Xiangyu Zhao, Shuofei Qiao, Minting Pan, Wangmeng Zuo, Lei Bai, Dongzhan Zhou, Ningyu Zhang, and Huajun Chen. LabVLA: Grounding vision-language-action models in scientific laboratories. *arXiv preprint arXiv:2606.13578*, 2026. URL https://arxiv.org/abs/2606.13578.

Cinjon Resnick, Roberta Raileanu, Sanyam Kapoor, Alexander Peysakhovich, Kyunghyun Cho, and Joan Bruna. Backplay: Man muss immer umkehren. *arXiv preprint arXiv:1807.06919*, 2018. URL https://arxiv.org/abs/1807.06919.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

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 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 258–282. PMLR, 2025. URL https://proceedings.mlr.press/v305/wagenmaker25a.html.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024. URL https://arxiv.org/abs/2311.12908.

Yihao Wu, He Zhang, Junbo Tan, Xueqian Wang, and Zhengyou Zhang. Flowpro: Reward-free reinforced fine-tuning of flow-matching vlas via proximalized preference optimization. *arXiv preprint arXiv:2606.05468*, 2026. URL https://arxiv.org/abs/2606.05468.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi Fan, Guanya Shi, and Yuke Zhu. Self-improving vision-language-action models with data generation via residual RL. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eUGoqrZ6Ea.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.16117.

APPENDIX

A ANALYSIS DETAILS

A.1 PROOF OF PROPOSITION 1

Fix the rooted decision (h, s) and keep the continuation policy at π_{ref}. For the terminal binary reward in equation 7, Bayes' rule gives

$$\pi^+(a) = \frac{Q(a)\pi_{\text{ref}}(a \mid s)}{V}, \qquad \pi^-(a) = \frac{[1 - Q(a)]\pi_{\text{ref}}(a \mid s)}{1 - V}. \tag{14}$$

These are distributions of the root action under the same normalized reference policy, root state, and subsequent decisions. They satisfy $\pi_{\text{ref}}(\cdot \mid s) = V\pi^+ + (1 - V)\pi^-$. Define

$$\psi(a) = \nabla_\theta \log \pi_\theta(a \mid s)|_{\theta=\theta_0} \tag{15}$$

and the one-step-deviation objective

$$\mathcal{J}(\theta) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[Q(a)], \qquad g := g_h(s) = \nabla\mathcal{J}(\theta_0). \tag{16}$$

The one-positive, K-negative empirical contrast is

$$\widehat{g}_K = \kappa \left(\psi(a^+) - \frac{1}{K}\sum_{k=1}^{K} \psi(a_k^-) \right), \qquad \kappa = V(1 - V). \tag{17}$$

Using the local shorthand from Section 4, score centering gives

$$\mathbb{E}_{a\sim\pi_{\text{ref}}}[\psi(a)] = 0. \tag{18}$$

Differentiating equation 16 at θ_0 yields

$$g = \mathbb{E}_{a\sim\pi_{\text{ref}}}[Q(a)\psi(a)]. \tag{19}$$

Using equation 14,

$$\mathbb{E}_{\pi^+}[\psi(a)] = \frac{g}{V}, \qquad \mathbb{E}_{\pi^-}[\psi(a)] = \frac{\mathbb{E}_{\pi_{\text{ref}}}[(1 - Q(a))\psi(a)]}{1 - V} = -\frac{g}{1 - V}. \tag{20}$$

Multiplying their difference by $V(1 - V) = \kappa$ proves equation 8; linearity of expectation then proves that equation 17 is unbiased.

Independence of the one positive and the K negative draws gives

$$\operatorname{Cov}\left[\psi(a^+) - \frac{1}{K}\sum_{k=1}^{K} \psi(a_k^-) \right] = \Sigma^+ + \frac{1}{K}\Sigma^-, \tag{21}$$

which proves equation 9. Consequently,

$$\mathbb{E}\|\widehat{g}_K\|_2^2 = \|g\|_2^2 + \kappa^2 \operatorname{tr}\left(\Sigma^+ + \frac{1}{K}\Sigma^- \right). \tag{22}$$

Finally, L-smoothness gives

$$\mathcal{J}(\theta_0 + \eta\widehat{g}_K) \geq \mathcal{J}(\theta_0) + \eta\langle g, \widehat{g}_K\rangle - \frac{\mathsf{L}\eta^2}{2}\|\widehat{g}_K\|_2^2. \tag{23}$$

Taking expectations and substituting unbiasedness and equation 22 gives

$$\begin{aligned} &\mathbb{E}[\mathcal{J}(\theta_0 + \eta\widehat{g}_K)] - \mathcal{J}(\theta_0) \\ &\geq \eta\|g\|_2^2 - \frac{\mathsf{L}\eta^2}{2}\left[\|g\|_2^2 + \kappa^2 \operatorname{tr}\left(\Sigma^+ + \frac{1}{K}\Sigma^- \right) \right]. \end{aligned} \tag{24}$$

Its right-hand side is positive for

$$0 < \eta < \frac{2\|g\|_2^2}{\mathsf{L}\left[\|g\|_2^2 + \kappa^2 \operatorname{tr}(\Sigma^+ + \Sigma^-/K)\right]}, \tag{25}$$

whenever $g \neq 0$.

Finally, $V = \mathcal{J}(\theta_0)$ is independent of the candidate action, so

$$\mathcal{J}(\theta) - \mathcal{J}(\theta_0) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[A_h^{\pi_{\text{ref}}}(s, a)]. \tag{26}$$

Thus improvement of the local objective is exactly positive expected reference-policy advantage for this one-step deviation.

A.2 Proof of Proposition 2

For the valid handoff-state distributions ν_j, the root gradients and their mean are

$$g_j = \mathbb{E}_{s\sim\nu_j}[g_{h_j}(s)], \qquad \bar{g} = \frac{1}{N}\sum_{j=1}^{N} g_j. \tag{27}$$

Consider root-specific copies $\Delta\theta_j$ of reference-policy coordinates with common symmetric positive-definite metric F and norm $\|\Delta\theta\|_F^2 = \Delta\theta^\top F\Delta\theta$. The root-averaged budget is

$$\frac{1}{N}\sum_{j=1}^{N}\|\Delta\theta_j\|_F^2 \le \delta^2. \tag{28}$$

The root-conditioned and tied first-order gain programs are

$$\Gamma_{\mathrm{root}} = \max_{\{\Delta\theta_j\}:\, N^{-1}\sum_{j=1}^{N}\|\Delta\theta_j\|_F^2\le\delta^2} \frac{1}{N}\sum_{j=1}^{N} g_j^\top\Delta\theta_j, \tag{29}$$

and, under $\Delta\theta_1 = \cdots = \Delta\theta_N = \Delta\theta$,

$$\Gamma_{\mathrm{pool}} = \max_{\|\Delta\theta\|_F\le\delta} \bar{g}^\top\Delta\theta. \tag{30}$$

Cauchy–Schwarz applied directly to the root-indexed corrections gives

$$\begin{aligned}\frac{1}{N}\sum_{j=1}^{N} g_j^\top\Delta\theta_j &\le \sqrt{\frac{1}{N}\sum_{j=1}^{N}\|g_j\|_{F^{-1}}^2}\sqrt{\frac{1}{N}\sum_{j=1}^{N}\|\Delta\theta_j\|_F^2} \\ &\le \delta\sqrt{\frac{1}{N}\sum_{j=1}^{N}\|g_j\|_{F^{-1}}^2}.\end{aligned} \tag{31}$$

When the denominator is nonzero, equality is attained by

$$\Delta\theta_j^\star = \frac{\delta F^{-1}g_j}{\sqrt{N^{-1}\sum_{i=1}^{N}\|g_i\|_{F^{-1}}^2}}, \tag{32}$$

which proves equation 10; the all-zero case is immediate. For the pooled problem, ordinary Cauchy–Schwarz gives

$$\bar{g}^\top\Delta\theta \le \|\bar{g}\|_{F^{-1}}\|\Delta\theta\|_F, \tag{33}$$

with equality at $\Delta\theta^\star = \delta F^{-1}\bar{g}/\|\bar{g}\|_{F^{-1}}$ whenever $\bar{g}\neq 0$. This proves equation 11.

Finally, expanding the pairwise distances yields

$$\begin{aligned}\frac{1}{2N^2}\sum_{i,j=1}^{N}\|g_i - g_j\|_{F^{-1}}^2 &= \frac{1}{N}\sum_{j=1}^{N}\|g_j\|_{F^{-1}}^2 - \|\bar{g}\|_{F^{-1}}^2 \\ &= \frac{1}{N}\sum_{j=1}^{N}\|g_j - \bar{g}\|_{F^{-1}}^2.\end{aligned} \tag{34}$$

Multiplying by δ^2 proves equation 12 and the claimed ordering.

A.3 Posterior mixtures in flow space

At the fixed context (h, s), equation 14 gives $\pi_{\mathrm{ref}} = V\pi^+ + (1-V)\pi^-$. Apply the same corruption process to each distribution. Let $v^\star_{\mathrm{ref}}$, $v^\star_+$ and $v^\star_-$ be their population-optimal squared-error flow fields. With $u_t = \epsilon - a$, conditional regression gives $v^\star_{\mathrm{ref}}(x_t, t \mid s) = \mathbb{E}[u_t \mid x_t, t, s]$. Conditioning additionally on Y and applying the law of total expectation yields

$$v^\star_{\mathrm{ref}} = \alpha v^\star_+ + (1-\alpha)v^\star_-, \tag{35}$$

where $\alpha = \Pr(Y = 1 \mid x_t, t, s)$ and the common field arguments are suppressed. Rearranging gives the positive–negative symmetry used by NFT (Zheng et al., 2026):

$$(1-\alpha)(v^\star_{\mathrm{ref}} - v^\star_-) = \alpha(v^\star_+ - v^\star_{\mathrm{ref}}). \tag{36}$$

The mixing weight is the success posterior at the corrupted action, not generally the prior success probability V.

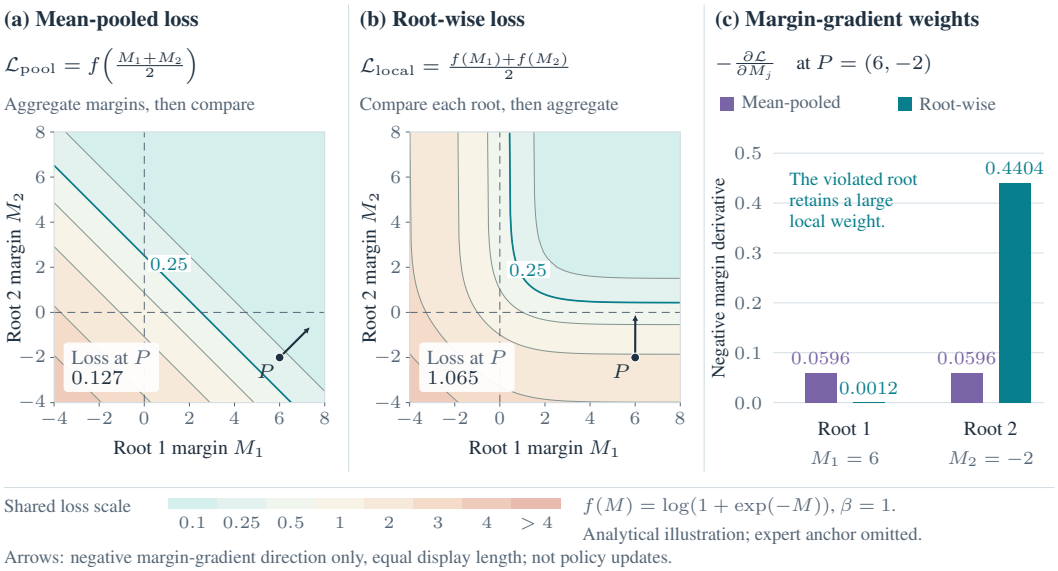


Figure 3: **Root-wise comparisons retain local optimization pressure.** Two roots, one pair each, $\beta = 1$, without expert attraction. **(a,b)** At $P = (6, -2)$, pooling hides a violated preference behind a positive margin; the root-wise loss retains its penalty. **(c)** Local negative margin derivatives downweight the satisfied comparison while retaining weight on the violated one. Arrows indicate directions in margin coordinates, not policy-parameter updates.

A.4 Loss aggregation and local preference weights

To extend the one-pair-per-root comparison in Section 4.2, define

$$\bar{M}_j = \frac{1}{K}\sum_{k=1}^{K} M_{jk}, \qquad \bar{M} = \frac{1}{N}\sum_{j=1}^{N} \bar{M}_j, \qquad f_\beta(M) = \text{softplus}(-\beta M). \tag{37}$$

Applying Jensen's inequality first within each root and then across roots gives

$$\mathcal{L}_{\text{pref}} \geq \mathcal{L}_{\text{root}} := \frac{1}{N}\sum_{j=1}^{N} f_\beta(\bar{M}_j) \geq f_\beta(\bar{M}) = \mathcal{L}_{\text{pool}}. \tag{38}$$

Furthermore, $\bar{M}_j \leq 0$ implies $f_\beta(\bar{M}_j) \geq f_\beta(0) = \log 2$, so

$$\frac{1}{N}\sum_{j=1}^{N} \mathbf{1}\{\bar{M}_j \leq 0\} \leq \frac{\mathcal{L}_{\text{root}}}{\log 2}. \tag{39}$$

Thus a small root-wise loss controls the fraction of roots with non-positive mean margin, while a small pooled loss constrains only their average. Figure 3 instantiates equation 38 for $N = 2$, $K = 1$, and $\beta = 1$: in that case $\bar{M}_j = M_{j1}$ and $\mathcal{L}_{\text{pref}} = \mathcal{L}_{\text{root}} = \mathcal{L}_{\text{local}}$. Since $f'_\beta(M) = -\beta\sigma(-\beta M)$, the chain rule gives equation 13; in particular, $M_j \leq 0$ implies a local derivative weight of at least $\beta/(2N)$. For multiple siblings, each individual pair instead has weight $\beta\sigma(-\beta M_{jk})/(NK)$ in $-\nabla_\theta \mathcal{L}_{\text{pref}}$. These are coefficients of the corresponding margin gradients, not independently realizable policy updates; Appendix B gives the energy-gradient form.

B Optimization and Collection Details

B.1 Query reconstruction and scoring

For RoboCasa, policy queries restart at each root every five control steps, with the final executed prefix truncated at the comparison-window boundary. Thus the resulting L_j queries need not lie on a

Table 4: **Representative root-access protocols.** Simulation entries are controller steps; real-robot entries are action frames. Values follow the stored root order.

Task	Root timestamps	Window lengths W_j	K
PastryDisplay (sim.)	$0, 70, 117, 131, 230, 284, 297, 313$	$131, 61, 14, 166, 67, 13, 37, 21$	8
Drawer storage (real)	$0, 258, 489, 601, 840, 1081, 1224$	$258, 231, 112, 239, 241, 143, 125$	10

global demonstration grid, and h_j denotes a rooted decision stage rather than a recorded timestamp. Scoring uses the native 50-step, 32-dimensional corruption and the 12 physical action dimensions at executed positions, with w_ℓ proportional to executed length. Replayed prefixes are excluded. Expert observations are rendered in the paired sibling's simulator context while reusing the demonstrated action reference.

B.2 Roots and comparison windows

The two protocols below illustrate one simulation and one physical instantiation. Both use task-specific semantic roots from one successful demonstration rather than a learned selector. PastryDisplay roots are reconstructed from expert gripper switches, placement events, and the success event; saved-state windows end at the next boundary in $(131, 297, 334)$ and may overlap. For Drawer storage, seven roots were manually marked in a 1,349-action-frame demonstration spanning two block placements and final drawer closure. Its windows end at the next root or trajectory end, and collection uses operator-assisted direct resets followed by phase confirmation rather than expert-prefix replay.

B.3 Flow-energy preference direction

Let $\phi(\tau) = \nabla_\theta E_\theta(\tau)|_{\theta_0}$. Since $M_{jk}(\theta_0) = 0$, differentiating the preference term in equation 5 gives

$$-\nabla_\theta \mathcal{P}_{jk}\big|_{\theta_0} = \frac{\beta}{2}\left[\phi(\tau_{jk}) - \phi(\tau_j^E)\right]. \tag{40}$$

Thus the initial preference direction combines expert-energy attraction with sibling-energy repulsion, while the explicit anchor adds $-\lambda\phi(\tau_j^E)$.

C Experimental Details

C.1 Training configuration

All simulation methods start from the official $\pi_{0.5}$ RoboCasa checkpoint and reuse its normalization statistics and action transform. BranchPO fine-tunes the action expert together with the action input/output projections and time multilayer perceptrons; the remaining pretrained parameters are frozen. The frozen initialization also serves as the reference and collection policy. Table 5 summarizes the optimization and segment-processing configuration.

Queries are reconstructed independently within each rooted window, and partial final queries are weighted by their executed length. Root-balanced sampling prevents longer comparison windows from receiving greater weight solely because they contain more control steps.

C.2 Compared post-training signals

SFT-only is the pretrained task policy before any experiment-specific update. One-demo SFT continues positive-only flow matching on the successful anchor demonstration and receives no autonomous sibling contrast. AWR performs return-weighted action regression on autonomous experience. Trajectory-DPO uses the same reference-relative flow-energy surrogate and expert-attraction term as BranchPO, but forms one preference margin for each full expert–autonomous trajectory pair. BranchPO instead forms phase-local margins between an expert segment and siblings collected from the same semantic root. None of these post-training comparisons requires new expert action labels beyond the available demonstration data.

Table 5: **BranchPO simulation post-training settings.** Settings are shared across the reported simulation tasks unless stated otherwise. Task-specific root and window definitions are listed in Table 4.

Setting	Configuration
Optimizer	AdamW
Peak learning rate	5×10^{-6}
Learning-rate schedule	4-step warmup; cosine decay to 5×10^{-7}
Weight decay	10^{-10}
Gradient-norm clipping	1.0
Exponential moving average	None
Preference parameters	$\beta = 1.0$; expert-attraction weight $\lambda = 0.1$
Pair sampling	Uniform over roots, then uniform over siblings within the selected root
Optimization batch	One root-local expert–sibling segment pair per update
Segment representation	48 local queries with masked padding
Native plan / scored dimensions	50 positions $\times$ 32 dimensions / 12 physical dimensions
Query cadence	One query every five control steps
Flow-time distribution	Inherited from pretraining
Corruption coupling	Shared flow time and noise across compatible current-policy and reference evaluations

Within each task, the methods share the observation and action interfaces, pretrained initialization, normalization, deployment controller, and official success predicate. The Trajectory-DPO scaling variants increase the number of from-start autonomous trajectories while retaining the full trajectory as their comparison unit.

C.3 Simulation evaluation setup

BranchPO collection accesses simulation roots through saved-state restoration, whereas every reported task evaluation begins from the full-task initial state. Each policy is evaluated on 50 initial configurations from a fixed task-specific set. Policies compared within a task share reset states, compiled scene identity, rollout seeds, and frozen flow noise. Evaluation varies robot and object poses while keeping the scene fixed, and success requires an exact Boolean `True` from the official `info.success` field.

C.4 Real-robot setup

The real-robot study evaluates banana placement, tissue retrieval, and drawer storage using 20 full-task trials per method and task. An operator returns the objects to approximately the same initial workspace region between trials, with natural variation in position and orientation. The SFT-only policy is the base used to initialize BranchPO, and its training set already contains the anchor demonstration. During BranchPO collection, the operator directly establishes the premarked semantic roots and confirms the task phase before autonomous control; evaluation itself always starts from the beginning of the task.

Controller timing, replanning cadence, blending, and speed limits are unchanged across the reported policies. Full-task success and the qualitative execution comparisons are assessed under the same task definitions.