AGENTIC-DPO: ROLLOUT-FREE AGENT TRAINING FROM THE STUDENT'S OWN MISTAKES

Anonymous authors
Paper under double-blind review

ABSTRACT

Expert trajectories are the cheapest supervision for LLM agents, yet supervised fine-tuning (SFT) never confronts the agent with the mistakes it would itself make at the expert's states. **Agentic-DPO** mines these mistakes without executing any action during training: at each expert state it samples a few one-step actions from the current student, keeps the most probable one whose decision differs from the expert's, and trains the student to prefer the expert action with a length-scaled DPO loss, stabilized by an SFT anchor and by **Policy-Preserving Augmentation** (PPA), which re-renders trajectories under different action schemas. We evaluate with ten seeds (five at 32B), validation-selected checkpoints, and twelve tuning configurations for every method. On five closed-loop tasks (τ-bench retail and airline with the standard GPT-4o user, ALFWorld, WebShop, and end-to-end StableToolBench), Agentic-DPO improves over SFT with the same augmentation by 5.9–15.2 points at 9B, matches or exceeds the step-level methods IPR and PivotRL, and recovers 93% (95% CI 77–111) of online GRPO's τ-bench gain with 6.7$\times$ fewer GPU-hours; GRPO started from Agentic-DPO has the highest mean on all five tasks. Matched controls show that the choice of negative matters more than the loss: on the same states, continued SFT recovers 21% of the gain and a random negative from the student 69%, whereas losses that keep pushing down the student's most probable mistake recover 88–97%. Its advantage over learning from the student's own samples comes mostly from states where the student never samples the expert's decision, where such methods receive no direct signal.

1 INTRODUCTION

LLM agents are increasingly trained to act: to call tools, talk to users, and navigate web pages (Qin et al., 2024; Guo et al., 2024; Yao et al., 2024; Deng et al., 2023). The workhorse for training open agents is supervised fine-tuning (SFT) on expert trajectories collected from stronger models, benchmark releases, or human annotators (Chen et al., 2023; Zeng et al., 2024; Liu et al., 2025; Chen et al., 2025). SFT is cheap, stable, and needs no environment, but its objective only imitates demonstrated actions. Consider a retail customer who gives a name and a zip code: the expert calls `find_user_id_by_name_zip`, while the student may be inclined to call `find_user_id_by_email`. SFT raises the probability of the expert's tokens and lowers every alternative a little, in proportion to its probability; it never looks at the complete call the student would actually make, so it never tells the student which of its own decisions is the mistake.

Reinforcement learning and preference learning provide this error-driven signal, but for agents they typically obtain it by executing the student's actions, training a reward model, or annotating failing steps (Song et al., 2024; Xiong et al., 2024; Lai et al., 2024; Da et al., 2025). A cheaper family samples single actions at the expert's states and rewards those that agree with the expert (Qian et al., 2025; Zhang et al., 2025; Yi et al., 2026); it needs no multi-turn rollouts, but its group-relative signal vanishes wherever none of the student's samples is right. Our observation is that at an expert state neither execution nor agreement is needed: the student's likely mistake can be *sampled* without being *executed*, and the expert action at the same state serves as the preferred alternative, so every state at which the student errs yields a training pair, including states at which it never samples the right decision.

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.

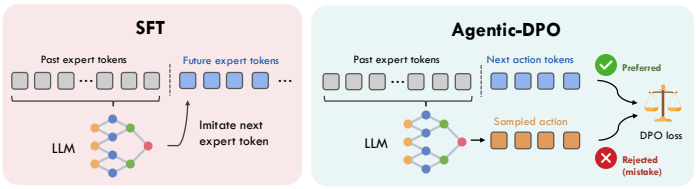

Figure 1: SFT versus Agentic-DPO. SFT imitates the expert tokens of a trajectory. Agentic-DPO treats each expert action as a decision at its state and trains the model to prefer it over the most probable different action sampled from the current student at the same state.

Agentic-DPO (Figure 1) turns this into a recipe. At each expert state the current student samples a few one-step actions; the most probable one whose decision differs from the expert's becomes the rejected action; the pair is optimized with a length-scaled DPO loss (Rafailov et al., 2023) and re-mined as the student improves. Two properties of agent actions shape the recipe. Many sampled "mistakes" are the expert's decision in another surface form, so candidates are parsed and compared at the level of decisions; and a rendered action entangles the decision with its schema, so we keep an SFT anchor and introduce **Policy-Preserving Augmentation** (PPA), which renders each trajectory under several action schemas and tool namings while keeping every expert decision fixed.

We evaluate on five closed-loop tasks and two open-loop benchmarks with four backbones from two model families, under a protocol designed to make comparisons fair: ten seeds (five at 32B), checkpoints selected on a validation split for every method, the same augmentation and twelve tuning configurations for every learned method, and confidence intervals over test tasks and seeds. Agentic-DPO improves the mean over SFT with the same augmentation everywhere, by 10.2–15.2 points of τ-bench success across backbones; it is on par with or ahead of step-level methods, including PivotRL; and at 9B it recovers 93% of online GRPO's τ-bench gain at 6.7$\times$ lower cost; none of its differences to GRPO on the five tasks (-1.4 to $+1.7$ points) is distinguishable from zero. GRPO started from Agentic-DPO has the highest mean on all five tasks.

Matched experiments explain why the recipe works; we believe the findings are useful beyond our method. (1) *The negative matters more than the loss.* On the same states and budget, training without a selected negative (continued SFT, step-level SPIN, a single filtered sample) recovers 21–30% of the τ-bench gain, as a local analysis predicts, and a random valid negative 69%; with the student's most probable mistake, every loss we tried that keeps pushing it down (IPO, CPO, hinge, SimPO, and a linear push-down) recovers 88–97% of the gain obtained with DPO. (2) *The expert gives a signal where the student's samples give none.* At held-out states where PPA+SFT never samples the expert's decision in eight tries, Agentic-DPO lowers the greedy error from 96% to 52%, whereas PivotRL and GRPO reach 90% and 78%. (3) *It complements online RL.* Agentic-DPO corrects decisions at the expert's states and GRPO at the states the student reaches on its own; each has the lowest error on its own kind of state, and their combination is best on both.

2 Related Work

Learning from demonstrations. SFT on expert trajectories is the standard recipe for open agents (Chen et al., 2023; Zeng et al., 2024; Chen et al., 2024b; Qin et al., 2024; Liu et al., 2025; Zeng et al., 2025; Fu et al., 2025), refined by function masking (Lin et al., 2024) and critical-step selection (Chen et al., 2025; Tao et al., 2026). These methods optimize supervised likelihood objectives, sometimes using trajectories that contain errors and subsequent corrections (Fu et al., 2025); they do not explicitly contrast the expert action with the current student's highest-probability non-equivalent action at the same expert state. Failed trajectories have also been added as SFT data with a failure tag (Wang et al., 2024). Imitation learning offers two classic remedies for the compounding errors of behavior cloning (Ross & Bagnell, 2010): querying an interactive expert at the student's states (DAgger (Ross et al., 2011), SEARN (Daumé III et al., 2009), LOLS (Chang et al., 2015)) and perturbing the demonstrator or synthesizing bad outcomes (DART (Laskey et al., 2017), ChauffeurNet (Bansal et al., 2019)). Agentic-DPO needs neither an interactive expert nor a simulator of the task: it queries the *student* at the expert's states. Max-margin planning (Ratliff et al., 2006), implicit behavioral cloning (Florence et al., 2022), and inverse soft-Q or distribution-matching methods (Garg et al.,

Table 1: Where the rejected sample comes from (citations in the text). Supervision: what training needs besides the demonstrations; Executes: whether training executes sampled actions in the environment.

Method	Unit	Rejected sample	Supervision	Executes
SFT, RFT	step	none	RFT: match check	no
SPIN, DITTO	full response	model's own sample	none	no
Step-DPO	reasoning step	step at a located error	error-step labels	no
ToolRL, Tool-N1	tool call	samples scored by match	match reward	no
PivotRL	turn at expert state	samples a verifier rejects	equivalence verifier	no
ETO	trajectory	failed exploration	outcome	yes
IPR	step	action scored by rollouts	outcome	yes
GRPO	trajectory group	on-policy rollouts	reward	yes
Agentic-DPO	decision at expert state	student's most probable other decision	decision parser	no

2021; Kostrikov et al., 2020; Wulfmeier et al., 2024) also contrast expert actions with alternatives, through a margin, an energy model, or a learned value.

Where the rejected sample comes from. Table 1 organizes the closest methods by the origin of the rejected sample. Online RL (Ouyang et al., 2022; Schulman et al., 2017; Shao et al., 2024; Qi et al., 2025; Bai et al., 2024; Wei et al., 2025; Wang et al., 2025b; Xi et al., 2025) and step-level methods such as IPR (Xiong et al., 2024), StepAgent (Deng et al., 2024), and Agent Q (Putta et al., 2024) execute the agent's actions to score them; ETO (Song et al., 2024) contrasts expert trajectories with failed exploration; TP-LLaMA (Chen et al., 2024a) pairs tool calls with failed branches of recorded search trees; Step-DPO (Lai et al., 2024) needs located error steps. Single-step RL at the expert's states rewards samples that match the expert's call (Qian et al., 2025; Zhang et al., 2025); PivotRL (Yi et al., 2026), the closest method to ours, rewards samples that a domain verifier accepts as equivalent to the demonstration and trains with group-relative advantages on the turns where the samples disagree. Like Agentic-DPO, it needs no multi-turn rollouts; unlike Agentic-DPO, it learns from the student's own samples only, so it has no signal at states where none of them is right. SPIN (Chen et al., 2024c; Wang et al., 2025a) and DITTO (Shaikh et al., 2024) contrast demonstrations with the model's own full responses and SLiC (Zhao et al., 2023) calibrates against ranked model samples; IRPO (Pang et al., 2024), RPO (Liu et al., 2024), and CPO (Xu et al., 2024) add a likelihood term on the preferred response; unlikelihood training (Welleck et al., 2020) and NPO (Zhang et al., 2024) push down unwanted outputs, and on-policy negative gradients are known to help preference fine-tuning (Tajwar et al., 2024). Agentic-DPO differs in *what* is contrasted: one decision at an expert state against the student's single most probable decision-different alternative, found by parsing and filtering candidates and re-mined as the student changes. Selecting the most violating output follows structured prediction (Collins, 2002), and the expected update of an unselected on-policy negative is the score-function identity behind contrastive divergence (Hinton, 2002). DPO (Rafailov et al., 2023) and its variants (Azar et al., 2024; Meng et al., 2024; Ethayarajh et al., 2024; Shi et al., 2024; Jung et al., 2025; Gao et al., 2025) supply the objective, and a likelihood anchor (Pal et al., 2024; Pang et al., 2024) counters the displacement of the chosen response (Razin et al., 2025).

3 METHOD

Agentic-DPO turns expert steps into preference pairs between the expert action and the student's most probable different decision at the same state, and optimizes them with a length-scaled DPO loss (Algorithm 1). Sections 3.1–3.3 describe the pair construction and the stabilizers, and Section 3.4 analyzes what the update does.

3.1 PROBLEM SETTING

The input is an expert trajectory dataset $\mathcal{D}_{\text{exp}} = \{\tau_i\}_{i=1}^{N}$ with $\tau_i = \{(s_{i,t}, u_{i,t}^{+})\}_{t=1}^{T_i}$, where $s_{i,t}$ is the agent state (instruction, history, available tools or actions, observations) and $u_{i,t}^{+}$ is the expert's *latent decision*, such as a tool with its arguments or a click on an element. A schema view $\phi \in \Phi$ renders states and decisions as text, $s_t^{\phi} = g_\phi(s_t)$ and $a^{\phi} = r_\phi(u)$; the same decision can be rendered, e.g., as a ReAct call or a JSON function call. The student $\pi_\theta(a \mid s)$ assigns probabilities to rendered action strings, and SFT minimizes $\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(s_t,u_t^{+}),\phi}\big[\log \pi_\theta(r_\phi(u_t^{+}) \mid g_\phi(s_t))\big]$, averaged over action

Algorithm 1 Agentic-DPO

Require: Expert trajectories $\mathcal{D}_{\text{exp}}$, schema views Φ, base policy π_{base}, refresh rounds R, updates per round M, batch size B, candidates K, coefficients β, α, λ

Warm up $\pi_\theta \leftarrow \pi_{\text{base}}$ with SFT on $\mathcal{D}_{\text{exp}}$ under online PPA; freeze $\pi_{\text{ref}} \leftarrow \pi_\theta$
for $r = 1, \ldots, R$ **do** ▷ Refresh round
  Snapshot $\theta_{\text{student}} \leftarrow \theta$; sample slice $\mathcal{S}_r \subset \mathcal{D}_{\text{exp}}$ of size $M \cdot B$; $\mathcal{D}_{\text{pref}}^{(r)} \leftarrow \emptyset$
  for each $(s_t, u_t^+) \in \mathcal{S}_r$ **do**
    $\phi \sim \text{Unif}(\Phi)$; draw K candidates from $\pi_{\theta_{\text{student}}}(\cdot \mid g_\phi(s_t))$ and parse them
    drop unparseable and decision-equivalent candidates; append (s_t, u_t^+, u_t^-) with u_t^- the most probable rest
  end for
  for M gradient steps **do** ▷ Update with PPA
    Minibatch from $\mathcal{D}_{\text{pref}}^{(r)}$; draw a fresh view $\psi \sim \text{Unif}(\Phi)$ per triple and render it
    Update θ with $\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{ADPO}} + \lambda \mathcal{L}_{\text{SFT}}$
  end for
end for
return π_θ

tokens. SFT raises the likelihood of the expert action and lowers every alternative in proportion to its probability; it never scores the complete action the student would produce. Agentic-DPO adds this contrast with triples (s_t, u_t^+, u_t^-) in which u_t^- is a one-step decision sampled from the current student at the same state. No sampled action is executed during training; only checkpoint selection on closed-loop benchmarks runs validation episodes (Section 4.1).

3.2 Agentic-DPO: State-Conditioned Action Preferences

At the beginning of each refresh round, the current student is the negative sampler. For an expert step (s_t, u_t^+), we draw a view ϕ, sample $K = 4$ candidates from $\pi_{\theta_{\text{student}}}(\cdot \mid s_t^\phi)$, and parse each into a decision $\tilde{u}^{(k)}$. Candidates that cannot be parsed, or whose decision equals the expert's (same tool and normalized arguments, same element and operation), are removed; if none remains, the step is skipped this round. Among the rest we take the one with the highest summed student log-probability, $u_t^- = \arg\max_k \log \pi_{\theta_{\text{student}}}(r_\phi(\tilde{u}^{(k)}) \mid s_t^\phi)$, i.e., the mistake the student is most likely to make. The decision filter matters: without it, 34% of the selected negatives are renderings of the expert's own decision (Section 4.4).

For training, each triple is rendered under a fresh view ψ (Algorithm 1), giving $a^\pm = r_\psi(u_t^\pm)$ and $s = g_\psi(s_t)$, and Agentic-DPO minimizes the length-scaled DPO loss

$$\mathcal{L}_{\text{ADPO}}(\theta) = -\mathbb{E}\left[\log \sigma\left(\beta_{\text{eff}}\left[\log \frac{\pi_\theta(a^+ \mid s)}{\pi_\theta(a^- \mid s)} - \log \frac{\pi_{\text{ref}}(a^+ \mid s)}{\pi_{\text{ref}}(a^- \mid s)}\right]\right)\right], \qquad \beta_{\text{eff}} = \frac{\beta}{\max(|a^+|, |a^-|)^\alpha}, \tag{1}$$

where log-probabilities are summed over the action tokens under ψ, $|a|$ is the number of action tokens, $\alpha = 0.5$, and π_{ref} is the frozen SFT-warmed student. The length scaling keeps long actions from dominating short ones; because β_{eff} depends on the pair, Eq. (1) is a DPO-style loss rather than the exact KL-regularized DPO objective. Both actions share the expert state, so the loss compares two decisions rather than two trajectories. In our runs β_{eff} is small enough that the loss stays in its near-linear regime (Section 3.4), where it acts as the anchor plus a push-down of $\log \pi_\theta(a^-)$; we keep the DPO form as a standard implementation, although at these settings the reference model does not affect the update, and Section 4.4 shows that the choice of negative matters more than the choice of loss.

3.3 Stabilizing Agentic-DPO

SFT warm-up and anchor. The warm-up (PPA+SFT, re-run for every seed) gives the student format validity and expert coverage before preference training; π_{ref} is frozen afterwards. During preference training we keep an SFT anchor, $\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{ADPO}} + \lambda \mathcal{L}_{\text{SFT}}$ with $\lambda = 0.5$, which keeps the expert action likely when the loss suppresses a negative that shares most of its tokens, a known failure mode of DPO (Razin et al., 2025).

Policy-Preserving Augmentation. PPA renders the same decision under several views, applied to the expert and the negative alike so that each pair stays schema-consistent. *Action-rendering views*, used on every benchmark, change how actions and tools are serialized and named (ReAct vs. JSON calls, consistently renamed tools; element and command renderings for web and text environments). *Recovery-context views*, used only in the tool domains (StableToolBench and τ-bench), insert an earlier side-effect-free call (a read-only lookup or search) with an error or empty observation before the current step, so the agent has recovered from a local mistake while the environment state and the current decision are unchanged; replaying such contexts in the τ-bench simulator leaves the database state unchanged (Appendix N). Action-rendering views are drawn uniformly at mining (ϕ) and at training (ψ); in the tool domains a recovery-context view is added independently with probability 0.25, and 24% of τ-bench pairs are mined under one. Our rationale is that under a single schema the log-ratio of a pair mixes the decision with schema-specific lexical features, such as the tokens of a tool name, so the loss can be lowered through either; varying the rendering keeps only the decision-level part of the signal consistent across updates. Section 4.4 tests each family, and Appendix N gives a recipe for new domains (a parser, one renderer per view, and a rename map).

3.4 Analysis: Which Negative Moves the Decision

We work in a latent-decision softmax model at one expert state s with logits z, expert decision a^* (one-hot δ_{a^*}), and reference π_{ref} equal to the policy at initialization; in this section a denotes a decision. The SFT logit gradient is $\pi_{\text{ref}}(\cdot \mid s) - \delta_{a^*}$. For a pair (a^*, a^-) and a constant β, the contrastive loss is $-\log \sigma(\beta \Delta_\theta)$ with the reference-subtracted margin $\Delta_\theta = [z_\theta(a^*) - z_\theta(a^-)] - [z_{\text{ref}}(a^*) - z_{\text{ref}}(a^-)]$. The two statements below are local (at initialization, in a tabular model) and are meant to predict which controls should work, not the size of the gains; Section 4.4 tests them.

Proposition 1 (Unselected negatives). *If $a^- \sim \tilde{q}(\cdot \mid s)$, the expected contrastive logit gradient at initialization is $\frac{\beta}{2}(\tilde{q} - \delta_{a^*})$ (a pair with $a^- = a^*$ contributes nothing). For an unfiltered on-policy sample ($\tilde{q} = \pi_{\text{ref}}$, allowing $a^- = a^*$) it equals $\frac{\beta}{2}$ times the SFT gradient, and for a single sample filtered to differ from a^* it equals $\frac{\beta}{2(1-\pi_{\text{ref}}(a^*|s))}$ times the SFT gradient.*

This is the score-function identity behind contrastive divergence (Hinton, 2002): a single on-policy negative, filtered or not, moves the logits in the SFT direction, so step-level self-contrast (SPIN-style) should add little to anchored SFT. Among single samples from the student, the direction changes only when the negative is *selected* among several candidates, which reweights $\tilde{q}$ toward particular competitors; the competitor that matters is the one that decides greedy decoding.

Proposition 2 (Selected negatives). *Let $a_1 = \arg\max_{a \neq a^*} z(a)$ be the strongest competitor (assumed unique) and $m = z(a^*) - z(a_1)$ the margin that decides greedy decoding. For a rule that never selects a^*, a gradient step of size η at initialization on the expected contrastive loss changes m by $\frac{\eta\beta}{2}(1 + \tilde{q}(a_1)) + O(\eta^2)$. With K candidates, the most probable valid candidate is a_1 whenever a_1 is sampled, so $\tilde{q}_{\text{mp}}(a_1) = q$, the probability that a_1 is among the candidates; the top-2 and random rules draw uniformly among the two most probable or all distinct valid candidates. The first-order gain is ordered most probable $\geq$ top-2 $\geq$ random, and the ratio of the most probable to the random rule, $(1+q)/(1+\tilde{q}_{\text{rand}}(a_1))$, is at most $2K/(K+1)$.*

Proofs are in Appendix L. Two properties of our runs connect the model to training. First, Agentic-DPO operates where DPO is nearly linear: $\beta_{\text{eff}}\Delta$ has median 0.021 and 95th percentile 0.083, so $\sigma(\cdot) \approx \frac{1}{2}$ and the loss acts as anchored SFT plus a push-down of $\log \pi_\theta(a^-)$ with weight $\beta_{\text{eff}}/2$. Second, although the push-down's coefficient is one to two orders of magnitude below the anchor's, the anchor's gradient is small after the warm-up and Adam normalizes each coordinate, so the contrastive term accounts for 38% of the update norm (Appendix F).

4 Experiments

4.1 Experimental Setup

Benchmarks. Five closed-loop tasks: τ-bench retail (Yao et al., 2024) (115 test tasks) and airline (50 tasks, zero-shot transfer from retail training), scored by pass[1] over four trials with the standard GPT-4o user simulator; ALFWorld (Shridhar et al., 2021) (134 unseen games) and WebShop (Yao et al.,

Table 2: Closed-loop suite with Qwen3.5-9B (validation-selected checkpoints; mean±std over ten seeds). τ-airline: zero-shot transfer from retail training; WebShop: average reward; SoPR: StableToolBench end-to-end solvable pass rate. GPU-h: total training cost on τ-retail, including warm-up, sampling or rollouts, and validation. **Bold**: best of the single-stage methods; †: 95% bootstrap interval (tasks and seeds) of the difference to the best includes zero. The last rows give differences with their 95% intervals. Llama-3.1-8B and Qwen3-32B: Appendix C.

Method	τ-retail	τ-airline	ALFWorld	WebShop	STB SoPR	GPU-h
PPA+SFT	$25.8_{\pm 2.1}$	$15.2_{\pm 2.2}$	$68.1_{\pm 2.0}$	$61.5_{\pm 1.7}$	$46.3_{\pm 1.1}$	6.3
RFT	$28.4_{\pm 1.8}$	$16.4_{\pm 2.1}$	$70.9_{\pm 3.3}$	$63.0_{\pm 2.1}$	$47.2_{\pm 1.5}$	16.0
DFT	$33.1_{\pm 2.1}$	$17.5_{\pm 2.6}$	$70.4_{\pm 2.5}$	$62.8_{\pm 1.6}$	$48.0_{\pm 1.9}$	8.8
ETO	$31.6_{\pm 2.2}$	$18.3_{\pm 3.2}$	$74.3_{\pm 2.8}$	$65.9_{\pm 1.8}$†	$49.4_{\pm 2.5}$	23.6
IPR	$36.0_{\pm 2.1}$	$20.6_{\pm 2.9}$†	$77.4_{\pm 2.4}$†	$66.8_{\pm 2.2}$†	$49.9_{\pm 2.0}$	53.6
PivotRL	$38.9_{\pm 2.2}$	$21.4_{\pm 2.7}$†	$77.0_{\pm 2.7}$†	$66.9_{\pm 1.7}$†	$51.0_{\pm 1.6}$	27.4
GRPO	**42.1**$_{\pm 1.7}$	$22.0_{\pm 2.5}$†	**79.6**$_{\pm 3.1}$	**68.3**$_{\pm 2.8}$	$51.5_{\pm 1.9}$†	112.2
Agentic-DPO	$41.0_{\pm 2.1}$†	**23.4**$_{\pm 3.2}$	$78.2_{\pm 1.7}$†	$67.4_{\pm 1.6}$†	**53.2**$_{\pm 1.1}$	16.7
Agentic-DPO → GRPO	$46.2_{\pm 2.6}$	$25.8_{\pm 2.5}$	$84.0_{\pm 1.8}$	$70.6_{\pm 1.7}$	$55.0_{\pm 1.8}$	70.4
Agentic-DPO − GRPO	−1.1 [−4.1, +1.8]	+1.4 [−2.4, +5.3]	−1.4 [−5.2, +2.3]	−0.9 [−3.5, +1.5]	+1.7 [−0.5, +3.9]	
Agentic-DPO − PivotRL	+2.1 [−1.0, +5.2]	+2.0 [−1.8, +6.0]	+1.2 [−2.2, +4.9]	+0.5 [−1.7, +2.7]	+2.2 [+0.1, +4.3]	
Ag.-DPO→GRPO − GRPO	+4.1 [+1.2, +7.2]	+3.8 [−0.5, +8.2]	+4.4 [+0.7, +8.4]	+2.3 [−0.3, +4.9]	+3.5 [+1.2, +5.9]	

2022) (average reward on 200 test instructions), with the expert trajectories and splits of ETO (Song et al., 2024); and StableToolBench (Guo et al., 2024), scored end to end by its solvable pass rate (SoPR, 765 queries). Two open-loop benchmarks: next-action accuracy on StableToolBench given the expert history, on the canonical test set and on a perturbed variant whose nine operators rewrite tool names, formats, descriptions, and prompts without changing the correct decision (Appendix M), and step success on Mind2Web (Deng et al., 2023), averaged over its three test splits, whose experts are human demonstrations. Interaction during training (ETO, IPR, GRPO) and validation on τ-bench use a Qwen3.5-9B user simulator, so the test user is never seen in training. Data and splits are listed in Appendix A.

Protocol. We hold out 10% of the training tasks of every benchmark for validation, and every method selects its checkpoint on this split (the refresh round for Agentic-DPO; an evaluation every 150 updates, including the initialization, for the baselines). Each configuration is run with ten seeds, each re-running the whole pipeline including the warm-up. Every method, Agentic-DPO included, receives the same tuning budget of twelve validation configurations per benchmark; Agentic-DPO's grid is centred on defaults ($\lambda = 0.5$, $\beta = 0.008$, $\alpha = 0.5$, $K = 4$, learning rate 2×10^{-5}, batch 16, $M = 150$, $R \leq 5$) that a pilot on StableToolBench validation had chosen before this study, and the search kept them on every benchmark (Appendix B); the baselines had no pilot. For closed-loop differences we report 95% intervals from a bootstrap that resamples test tasks, paired across methods, and seeds; open-loop comparisons use Welch's test over seeds; Holm correction is applied within pre-specified families (Appendix D).

Baselines. All learned baselines start from the PPA+SFT warm-up and train on the same PPA views. RFT (Yuan et al., 2023) fine-tunes on the student's samples that match the expert decision; DFT (Wu et al., 2025) is a reward-rectified SFT; ETO (Song et al., 2024) is trajectory-level DPO against failed explorations, with our SFT anchor; IPR (Xiong et al., 2024) scores steps by Monte-Carlo rollouts; PivotRL (Yi et al., 2026) runs group-relative RL at expert states with our decision-equivalence check as its verifier and its own selection of informative turns; GRPO (Shao et al., 2024) ($G = 8$) trains end to end on task success, on StableToolBench with the SoPR evaluator's judgment. IPR and GRPO need an environment and are not run on static Mind2Web. Backbones are Qwen3.5-2B/4B/9B (Qwen Team, 2026) (Hugging Face `Qwen/Qwen3.5-2B`, `-4B`, `-9B`; revisions in the reproducibility statement), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Gemma3-4B (Gemma Team, 2025) for the stabilizer ablation, and Qwen3-32B (Yang et al., 2025) for scale (five seeds).

4.2 Closed-Loop Results

Agentic-DPO recovers most of online RL's gain at a fraction of its compute (Table 2). It improves over PPA+SFT by 5.9 to 15.2 points on the five tasks, over the best offline baseline (RFT, DFT, or ETO) by 1.5 to 7.9, and over IPR, which scores steps by rollouts at 3.2× its cost, by 0.6 to 5.0. It has a higher mean than PivotRL on every task (by 0.5 to 2.2 points; interval above zero on 1 of five) at

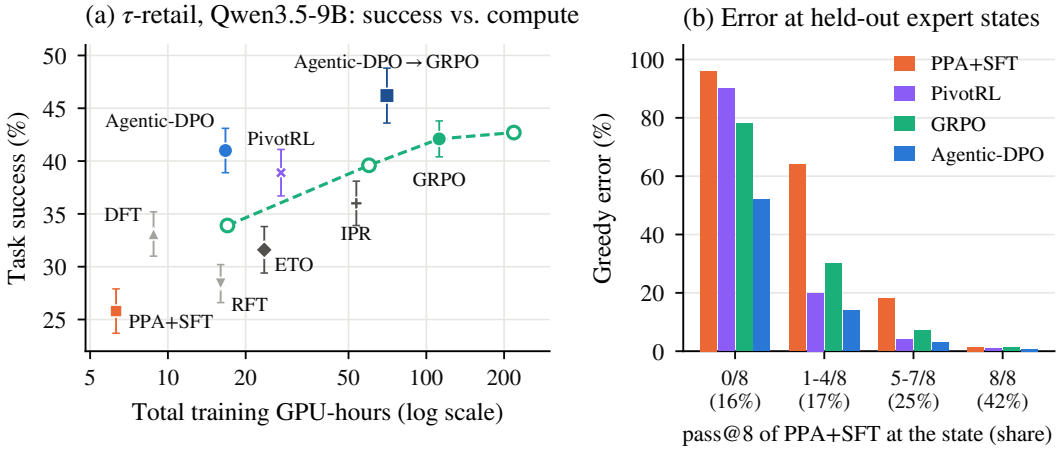

Figure 2: **(a)** τ-retail with Qwen3.5-9B: test success (mean$\pm$std over ten seeds) against total training GPU-hours. Hollow circles: GRPO re-tuned for 17 and 60 GPU-hours and trained for twice as many updates. **(b)** Greedy error at held-out τ-bench expert states, grouped by how often the PPA+SFT student samples the expert's decision in eight tries (share of states in parentheses).

1.6$\times$ lower cost. Against GRPO, the means differ by -1.4 to +1.7 points, GRPO ahead on three tasks and Agentic-DPO on airline transfer and SoPR, and every interval includes zero; the intervals exclude differences beyond about five points, so we read this as close to online RL rather than equivalent to it. Agentic-DPO recovers 93% (95% interval 77–111%) of GRPO's gain over PPA+SFT on τ-retail, 88% on ALFWorld, and 87% on WebShop, with 16.7 instead of 112.2 GPU-hours on τ-retail (5.9–6.7$\times$ less across tasks). GRPO re-tuned for Agentic-DPO's budget reaches 33.9% (Figure 2a).

Agentic-DPO is a good starting point for RL. GRPO started from Agentic-DPO has the highest mean on all five tasks, 2.3 to 4.4 points above GRPO from PPA+SFT (interval above zero on 3 of five), with fewer GPU-hours in total (70.4 vs. 112.2; its GRPO stage uses half as many updates). On τ-retail, starting GRPO from DFT, ETO, or PivotRL gives lower means (43.0, 43.3, and 44.0% vs. 46.2%), although the differences to the ETO and PivotRL starts are within noise (Table 31).

4.3 Across Backbones and Open-Loop Metrics

Table 3 repeats the comparison with Qwen3.5-2B/4B and Llama-3.1-8B. Agentic-DPO is best or not significantly different from the best in all 16 backbone–benchmark cells, improves τ-retail success over PPA+SFT by 10.2 to 15.2 points on every backbone (Holm-adjusted $p < 0.001$), and stays within -1.9 to -0.2 points of GRPO. On Mind2Web, whose experts are human, it improves step success by 8.7 to 10.9 points over PPA+SFT and by 1.5 to 3.8 over the best baseline. Open-loop StableToolBench is close to saturation at 9B, where the methods trained on PPA are within about one point of each other. With Qwen3-32B the gain over PPA+SFT shrinks but stays large (7.6 τ-retail points), GRPO leads by 2.4, and GRPO started from Agentic-DPO has the highest mean on all five tasks (Appendix C).

4.4 Where the Gain Comes From

Finding 1: the negative matters. Training without a selected negative recovers little of the τ-retail gain (Table 4a): continued SFT on the mined states, which isolates the state selection that mining performs, 21%; step-level SPIN 24%; and a single filtered sample 30%, consistent with Proposition 1, by which its expected update is the SFT direction; per pair it reaches only half of the first-order margin gain, a shortfall that the first-order model does not explain (Appendix F). Off-policy negatives, from the untrained base model or rule-based perturbations of the expert action, recover 30–36%. Selection among the student's own candidates is what helps: a random valid candidate recovers 69%, one of the two most probable 86%, and the most probable all of it, the order of Proposition 2; the most probable negative beats the random one on all three benchmarks after Holm correction over the 62 tests of the mechanism study (Appendix D). In the first round, the most probable, top-2, and random rules pick the greedy competitor in 0.77, 0.47, and 0.43 of their τ-bench pairs, and the predicted

Table 3: Four backbones (mean±std over ten seeds). τ-retail and SoPR: closed loop, as in Table 2; STB perturbed: next-action accuracy on perturbed StableToolBench; Mind2Web: step success. Bold and † as in Table 2 (Welch's test for the open-loop columns). All methods and the canonical StableToolBench column: Appendix C.

Method	τ-retail	STB SoPR	STB perturbed	Mind2Web
Qwen3.5-2B				
PPA+SFT	$21.1_{\pm 2.5}$	$33.4_{\pm 1.1}$	$81.4_{\pm 2.6}$	$41.6_{\pm 2.3}$
ETO	$28.5_{\pm 2.1}$	$35.1_{\pm 1.9}$	$\mathbf{85.0}_{\pm 1.9}$	$48.3_{\pm 2.2}$
PivotRL	$33.0_{\pm 2.2}$†	$37.3_{\pm 1.2}$	$84.9_{\pm 1.9}$†	$48.2_{\pm 2.1}$
GRPO	$\mathbf{34.5}_{\pm 1.4}$	$37.7_{\pm 2.0}$	$83.8_{\pm 2.5}$†	–
Agentic-DPO	$33.3_{\pm 2.6}$†	$\mathbf{41.3}_{\pm 1.8}$	$\mathbf{85.0}_{\pm 2.0}$	$\mathbf{51.0}_{\pm 2.2}$
Qwen3.5-4B				
PPA+SFT	$27.0_{\pm 2.0}$	$37.4_{\pm 1.7}$	$78.1_{\pm 3.2}$	$46.6_{\pm 2.6}$
ETO	$33.1_{\pm 2.6}$	$41.1_{\pm 2.1}$	$86.6_{\pm 1.7}$	$52.2_{\pm 2.0}$
PivotRL	$35.4_{\pm 1.7}$	$42.9_{\pm 1.9}$	$86.6_{\pm 1.8}$	$52.9_{\pm 2.1}$
GRPO	$\mathbf{39.1}_{\pm 2.2}$	$42.0_{\pm 1.4}$	$85.4_{\pm 2.6}$	–
Agentic-DPO	$37.2_{\pm 1.9}$†	$\mathbf{45.4}_{\pm 1.7}$	$\mathbf{88.4}_{\pm 1.9}$	$\mathbf{55.3}_{\pm 2.6}$
Qwen3.5-9B				
PPA+SFT	$25.8_{\pm 2.1}$	$46.3_{\pm 1.1}$	$90.3_{\pm 1.2}$	$52.7_{\pm 2.0}$
ETO	$31.6_{\pm 2.2}$	$49.4_{\pm 2.5}$	$91.2_{\pm 1.6}$†	$59.0_{\pm 1.9}$
PivotRL	$38.9_{\pm 2.2}$	$51.0_{\pm 1.6}$	$91.1_{\pm 1.4}$†	$59.8_{\pm 1.8}$
GRPO	$\mathbf{42.1}_{\pm 1.7}$	$51.5_{\pm 1.9}$†	$90.6_{\pm 1.6}$†	–
Agentic-DPO	$41.0_{\pm 2.1}$†	$\mathbf{53.2}_{\pm 1.1}$	$\mathbf{91.5}_{\pm 1.3}$	$\mathbf{63.6}_{\pm 1.8}$
Llama-3.1-8B				
PPA+SFT	$23.6_{\pm 1.7}$	$42.0_{\pm 1.1}$	$87.6_{\pm 2.0}$†	$49.0_{\pm 2.3}$
ETO	$27.8_{\pm 2.0}$	$47.5_{\pm 2.1}$	$\mathbf{89.3}_{\pm 1.8}$	$54.5_{\pm 2.3}$
PivotRL	$33.8_{\pm 2.2}$	$50.1_{\pm 1.1}$†	$88.8_{\pm 1.8}$†	$56.0_{\pm 2.1}$
GRPO	$\mathbf{37.9}_{\pm 1.5}$	$47.4_{\pm 2.0}$	$87.5_{\pm 2.3}$†	–
Agentic-DPO	$37.7_{\pm 2.2}$†	$\mathbf{50.5}_{\pm 1.6}$	$88.7_{\pm 1.7}$†	$\mathbf{59.6}_{\pm 2.0}$

first-order ratio of per-pair margin gains between the most probable and the random rule, 1.24, is close to the measured 1.26 (Appendix F); in the trained model, the mined negative is the greedy decision at 83–88% of the states where greedy decoding errs.

Finding 2: the loss matters less. With the most probable negative, losses that keep pushing it down recover 88–97% of the gain (IPO, CPO, hinge, SimPO, and a linear push-down; Table 4b and Appendix E), none significantly below DPO; in the near-linear regime of Section 3.4, DPO is itself this push-down plus the anchor. Losses whose push stops early recover less: unlikelihood, whose gradient vanishes once the negative is unlikely, 57%, the only loss significantly below DPO after Holm correction, and KTO 78% (a point estimate). Group-relative RL on the same four samples, the update of PivotRL, recovers 80% (also a point estimate): its advantage is zero at states where no sample is right, which is where Figure 2b locates most of Agentic-DPO's advantage over PivotRL.

Finding 3: the unit, the filter, and the stabilizers. Trajectory-level pairs trained with Agentic-DPO's settings recover 42% (random failure) and 50% (most probable failure), because one comparison is spread over many steps that mostly agree (Table 4c). Without the decision filter a third of the pairs contrast renderings of the expert's own decision, and the share falls to 43%; mining once instead of refreshing costs 5.5 points. Without the anchor, mean accuracy falls below PPA+SFT on every benchmark, consistent with likelihood displacement (Razin et al., 2025). Recovery-context views matter in closed loop: without them 71% of the gain remains (significant before, not after, Holm correction), and inserting the same calls with successful, irrelevant observations instead of errors recovers only 74%.

4.5 Where Agentic-DPO Helps, and Where RL Still Does

Figure 2b groups held-out τ-bench expert states by how often the PPA+SFT student samples the expert's decision in eight tries. At the 16% of states where it never does, the greedy error falls from 96% to 52% with Agentic-DPO but only to 90% with PivotRL, whose signal needs a correct sample, and 78% with GRPO; over all states the error falls from 31.4% to 11.8% (PivotRL 19.2%, GRPO 19.8%). The picture reverses at the states the student reaches on its own: at states from its own test episodes, labelled by a judge validated against human labels (Appendix G), GRPO has the lowest

Table 4: Where the gain comes from (validation-selected; mean±std over ten seeds): StableToolBench next-action accuracy with Qwen3.5-2B, τ-retail success and Mind2Web step success with Qwen3.5-9B. Every row starts from PPA+SFT and uses the same PPA views, number of updates, and checkpoint selection; rows in (a) and (b) train on the states mined by Agentic-DPO (re-mining with their own student: Appendix E), and each control's own weight is tuned on validation. Last column: share of Agentic-DPO's τ-retail gain over PPA+SFT recovered, with a 95% bootstrap interval. Recovery-context views are not used on Mind2Web.

Variant	StableToolBench (2B)		τ-retail	Mind2Web	Share of
	Canonical	Perturbed	(9B)	(9B)	τ gain (%)
PPA+SFT (starting point of every row)	$83.7_{\pm 2.3}$	$81.4_{\pm 2.6}$	$25.8_{\pm 2.1}$	$52.7_{\pm 2.0}$	0
(a) Which negative (DPO loss with anchor, same mined states)					
no negative: continued SFT on the mined states	$85.3_{\pm 2.0}$	$82.8_{\pm 2.2}$	$29.0_{\pm 1.5}$	$54.5_{\pm 1.8}$	21 [2, 38]
one on-policy sample, no decision filter (step-level SPIN)	$85.1_{\pm 2.0}$	$82.4_{\pm 2.3}$	$29.4_{\pm 1.3}$	$55.9_{\pm 1.9}$	24 [7, 39]
one filtered on-policy sample ($K = 1$)	$86.1_{\pm 1.8}$	$82.9_{\pm 2.0}$	$30.4_{\pm 1.8}$	$56.3_{\pm 1.8}$	30 [13, 47]
sample from the untrained base model	$86.8_{\pm 1.9}$	$82.5_{\pm 2.2}$	$31.3_{\pm 1.9}$	$57.2_{\pm 1.9}$	36 [18, 53]
rule-perturbed expert action	$86.0_{\pm 1.9}$	$81.9_{\pm 2.2}$	$30.4_{\pm 2.3}$	$56.4_{\pm 1.9}$	30 [12, 48]
random valid sample among $K = 4$	$87.0_{\pm 1.8}$	$82.2_{\pm 2.1}$	$36.3_{\pm 1.7}$	$60.3_{\pm 1.6}$	69 [53, 87]
one of the two most probable	$89.0_{\pm 1.5}$	$83.8_{\pm 1.9}$	$38.9_{\pm 2.1}$	$62.0_{\pm 1.5}$	86 [68, 106]
most probable valid sample (Agentic-DPO)	$\mathbf{90.3}_{\pm 1.7}$	$\mathbf{85.0}_{\pm 2.0}$	$\mathbf{41.0}_{\pm 2.1}$	$\mathbf{63.6}_{\pm 1.8}$	100
(b) Which loss (most probable negative)					
unlikelihood on the negative	$87.6_{\pm 1.8}$	$83.7_{\pm 2.1}$	$34.5_{\pm 1.9}$	$58.6_{\pm 1.7}$	57 [41, 74]
KTO (unpaired)	$88.9_{\pm 1.6}$	$84.1_{\pm 1.9}$	$37.7_{\pm 2.4}$	$61.1_{\pm 1.6}$	78 [60, 98]
group-relative RL on the same K samples	$88.6_{\pm 1.7}$	$84.0_{\pm 2.0}$	$38.0_{\pm 1.9}$	$61.0_{\pm 1.7}$	80 [63, 100]
IPO	$89.6_{\pm 1.5}$	$84.6_{\pm 1.7}$	$40.3_{\pm 2.0}$	$62.7_{\pm 1.5}$	95 [77, 117]
CPO (no reference model)	$89.8_{\pm 1.6}$	$84.6_{\pm 1.9}$	$40.4_{\pm 2.3}$	$62.9_{\pm 1.7}$	96 [78, 117]
linear push-down: anchor $+ c \log \pi_\theta(a^-)$	$89.9_{\pm 1.6}$	$84.8_{\pm 1.9}$	$40.6_{\pm 2.3}$	$63.1_{\pm 1.8}$	97 [79, 117]
(c) Pairing unit, filter, and stabilizers					
trajectory pairs, random failure	$87.8_{\pm 1.7}$	$85.1_{\pm 1.9}$	$32.2_{\pm 2.4}$	$59.3_{\pm 1.9}$	42 [24, 60]
trajectory pairs, most probable failure	$88.2_{\pm 1.6}$	$85.3_{\pm 1.8}$	$33.4_{\pm 2.5}$	$59.9_{\pm 1.8}$	50 [32, 68]
no decision-equivalence filter	$84.9_{\pm 2.1}$	$80.6_{\pm 2.5}$	$32.3_{\pm 1.9}$	$57.6_{\pm 1.9}$	43 [27, 59]
mine once ($R = 1$, same number of updates)	$88.0_{\pm 1.7}$	$83.9_{\pm 1.9}$	$35.5_{\pm 2.0}$	$60.6_{\pm 1.7}$	64 [47, 82]
without the SFT anchor	$72.1_{\pm 4.0}$	$67.9_{\pm 4.3}$	$24.3_{\pm 2.0}$	$50.2_{\pm 3.1}$	−10 [−32, 7]
without recovery-context views	$88.8_{\pm 1.8}$	$80.9_{\pm 2.4}$	$36.6_{\pm 1.9}$	–	71 [54, 90]

error (16.1%, Agentic-DPO 19.8%, PPA+SFT 27.9%), and it keeps an edge on the longest tasks (-6.2 points for Agentic-DPO minus GRPO on tasks with ten or more expert actions, 95% interval $[-12.1, -1.3]$). Starting GRPO from Agentic-DPO lowers both errors (10.3% and 13.5%). The mined negatives are genuine mistakes: in a blind audit of 600 negatives per benchmark from three rounds and two seeds (Appendix H), 84.2–90.2% are wrong actions and 6.2–10.0% valid alternatives, mostly rephrased replies on τ-bench; removing pairs whose negative may be a valid alternative changes success by at most 0.8 points.

5 Cost, Scope, and Limitations

Cost. Per update, Agentic-DPO costs 1.6× the wall time of SFT, PivotRL 3.9×, and GRPO 13.6× with $G = 4$ and 25.8× with $G = 8$ (end-to-end GPU-hours: Table 2 and Appendix I).

Limitations. Agentic-DPO learns at the expert's states and at the recovery contexts synthesized around them, so it cannot correct errors at states the student reaches on its own; online RL does, and we see Agentic-DPO as its initialization rather than its replacement. It skips states without a different valid candidate (21.2% to 33.0% of τ-bench states over the rounds), needs a decision parser per action space, and trusts the expert: with 20% corrupted expert actions it loses 5.4 τ-retail points (Appendix K). Where several actions are valid, some negatives are acceptable alternatives. We evaluate up to 32B parameters, with simulated users and cached tools.

6 Conclusion

Expert trajectories contain more supervision than SFT extracts. By pushing down the student's own mistakes at expert states while anchoring the expert, Agentic-DPO turns demonstrations into positive and negative signal without executing an action during training; it recovers most of online RL's gain at a fraction of its compute and gives online RL a better starting point.

Reproducibility statement

The training and evaluation code, the PPA transformations, the validation splits and seeds, the StableToolBench-Perturbed generation script, the negative-audit guidelines and labels, the per-seed and per-task results of every table, and the configuration files for every method and experiment are included in the supplementary material. Appendix A lists data sources, splits, and how the τ-bench expert dialogues were produced; Appendix B gives implementation details and the tuning protocol of every method; Appendix D the statistical procedures; Appendix I the full compute accounting; Appendix M the perturbation operators and the fixed seed of the evaluation set; and Appendix L the assumptions and proofs of the analysis.

The Qwen3.5 backbones are the Hugging Face checkpoints `Qwen/Qwen3.5-2B` at revision `15852e8c16360a2fea060d615a32b45270f8a8fc`, `Qwen/Qwen3.5-4B` at revision `851bf6e806efd8d0a36b00ddf55e13ccb7b8cd0a`, and `Qwen/Qwen3.5-9B` at revision `c202236235762e1c871ad0ccb60c8ee5ba337b9a`; the 9B checkpoint also serves as the training and validation user simulator of τ-bench.

Ethics statement

This work trains and evaluates language-model agents on publicly released benchmarks and does not involve private data. The negative audit was carried out by two annotators on benchmark actions only. The method lowers the cost of agent training and inherits the behavior of the expert trajectories it is trained on, so a systematic error or bias in the expert data would be reinforced (Appendix K).

AI use statement

We used large language model tools to edit the text of this manuscript, to help draft parts of the analysis (Section 3.4) and the statistical appendix, to write plotting scripts, as the user simulators of τ-bench, and as the LLM judges named in the appendix. These tools were not used to generate experimental results. The authors reviewed all AI-assisted text and take full responsibility for the content of this paper.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. DigiRL: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 37:12461–12495, 2024.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. In *Robotics: Science and Systems (RSS)*, 2019.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daumé III, and John Langford. Learning to search better than your teacher. In *International Conference on Machine Learning (ICML)*, 2015.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. FireAct: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023.

Sijia Chen, Yibo Wang, Yi-Feng Wu, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. Advancing tool-augmented large language models: Integrating insights from errors in inference trees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-FLAN: Designing data and methods of effective agent tuning for large language models. *arXiv preprint arXiv:2403.12881*, 2024b.

Zhixun Chen, Ming Li, Yuxuan Huang, Yali Du, Meng Fang, and Tianyi Zhou. ATLaS: Agent tuning via learning critical steps. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning (ICML)*, 2024c.

Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

Jeff Da, Clinton Wang, Xiang Deng, Yuntao Ma, Nikhil Barhate, and Sean Hendryx. Agent-RLVR: Training software engineering agents via guidance and environment rewards. *arXiv preprint arXiv:2506.11425*, 2025.

Hal Daumé III, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine Learning*, 75(3):297–325, 2009.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, 2023.

Zhirui Deng, Zhicheng Dou, Yutao Zhu, Ji-Rong Wen, et al. From novice to expert: LLM agent policy optimization via step-wise reinforcement learning. *arXiv preprint arXiv:2411.03817*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning (ICML)*, 2024.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 158–168. PMLR, 2022. URL `https://proceedings.mlr.press/v164/florence22a.html`.

Dayuan Fu, Keqing He, Yejie Wang, Wentao Hong, Zhuoma Gongque, Weihao Zeng, Wei Wang, Jingang Wang, Xunliang Cai, and Weiran Xu. AgentRefine: Enhancing agent generalization through refinement tuning. In *International Conference on Learning Representations (ICLR)*, 2025.

Heyang Gao, Zexu Sun, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Xu Chen. Solving the granularity mismatch: Hierarchical preference learning for long-horizon LLM agents. *arXiv preprint arXiv:2510.03253*, 2025.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. IQ-Learn: Inverse soft-Q learning for imitation. In *Advances in Neural Information Processing Systems*, 2021.

Gemma Team. Gemma 3 technical report. Technical report, Google DeepMind, 2025. arXiv:2503.19786.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Sunghee Jung, Donghun Lee, Shinbok Lee, Gaeun Seo, Daniel Lee, Byeongil Ko, Junrae Cho, Kihyun Kim, EungGyun Kim, and Myeongcheol Shin. DiaTool-DPO: Multi-turn direct preference optimization for tool-augmented large language models. In Frédéric Béchet, Fabrice Lefèvre, Nicholas Asher, Seokhwan Kim, and Teva Merlin (eds.), *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 397–416, Avignon, France, August 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.sigdial-1.32. URL https://aclanthology.org/2025.sigdial-1.32/.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations (ICLR)*, 2020.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-DPO: Step-wise preference optimization for long-chain reasoning of LLMs. *arXiv preprint arXiv:2406.18629*, 2024.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise injection for robust imitation learning. In *Conference on Robot Learning (CoRL)*, 2017.

Qiqiang Lin, Muning Wen, Qiuying Peng, Guanyu Nie, Junwei Liao, Jun Wang, Xiaoyun Mo, Jiamu Zhou, Cheng Cheng, Yin Zhao, Jun Wang, and Weinan Zhang. Hammer: Robust function-calling for on-device language models via function masking. *arXiv preprint arXiv:2410.04587*, 2024.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. Provably mitigating overoptimization in RLHF: Your SFT loss is implicitly an adversarial regularizer. *arXiv preprint arXiv:2405.16436*, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *arXiv preprint arXiv:2402.13228*, 2024.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Manoj Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. APIGen-MT: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent Q: Advanced reasoning and learning for autonomous AI agents. *arXiv preprint arXiv:2408.07199*, 2024.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *The Thirteenth International Conference on Learning Representations (ICLR)*,

2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c66e1fcc9691aae706250638f36f681b-Abstract-Conference.html.

Cheng Qian, Emre Can Acikgoz, et al. ToolRL: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations (ICLR)*, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Maximum margin planning. In *International Conference on Machine Learning (ICML)*, 2006.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *International Conference on Learning Representations (ICLR)*, 2025.

Stéphane Ross and J. Andrew Bagnell. Efficient reductions for imitation learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Omar Shaikh, Michelle S. Lam, Joey Hejna, Yijia Shao, Michael S. Bernstein, and Diyi Yang. Show, don't tell: Aligning language models with demonstrated feedback. *arXiv preprint arXiv:2406.00888*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. Direct multi-turn preference optimization for language agents. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization for LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of LLMs should leverage suboptimal, on-policy data. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47441–47474. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/tajwar24a.html.

Chaofan Tao, Jierun Chen, Yuxin Jiang, Kaiqi Kou, Shaowei Wang, Ruoyu Wang, Xiaohui Li, Sidi Yang, Yiming Du, Jianbo Dai, et al. SWE-Lego: Pushing the limits of supervised fine-tuning for software issue resolving. *arXiv preprint arXiv:2601.01426*, 2026.

Renxi Wang, Haonan Li, Xudong Han, Yixuan Zhang, and Timothy Baldwin. Learning from failure: Integrating negative examples when fine-tuning large language models as agents. *arXiv preprint arXiv:2402.11651*, 2024.

Yibo Wang, Hai-Long Sun, Guangda Huzhang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. Triplets better than pairs: Towards stable and effective self-play fine-tuning for LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025b.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. WebAgent-R1: Training web agents via end-to-end multi-turn reinforcement learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7909–7928, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.401. URL https://aclanthology.org/2025.emnlp-main.401/.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations (ICLR)*, 2020.

Yongliang Wu, Yizhou Zhou, Ziheng Zhou, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of SFT: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*, 2025.

Markus Wulfmeier, Michael Bloesch, Nino Vieillard, Arun Ahuja, Jorg Bornschein, Sandy Huang, Artem Sokolov, Matt Barnes, Guillaume Desjardins, Alex Bewley, et al. Imitating language via scalable inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, et al. AgentGym-RL: Training LLM agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*, 2025.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Watch every step! LLM agent learning via iterative step-level process refinement. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *International Conference on Machine Learning (ICML)*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Junkeun Yi, Damon Mosk-Aoyama, Baihe Huang, Ritu Gala, Charles Wang, Sugam Dipak Devare, Khushi Bhardwaj, Abhibha Gupta, Oleksii Kuchaiev, Jiantao Jiao, et al. PivotRL: High accuracy agentic post-training at low compute cost. *arXiv preprint arXiv:2603.21383*, 2026.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

Xingshan Zeng, Weiwen Liu, Xu Huang, Zezhong Wang, Lingzhi Wang, Liangyou Li, Yasheng Wang, Lifeng Shang, Xin Jiang, Ruiming Tang, and Qun Liu. ToolACE-R: Model-aware iterative training and adaptive refinement for tool learning. *arXiv preprint arXiv:2504.01400*, 2025.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *Conference on Language Modeling (COLM)*, 2024.

Shaokun Zhang et al. Nemotron-Research-Tool-N1: Exploring tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*, 2025.

Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J. Liu. Calibrating sequence likelihood improves conditional language generation. In *International Conference on Learning Representations (ICLR)*, 2023.

A Data, Splits, and Expert Sources

Table 5: Benchmarks, expert sources, and splits. The validation split (10% of the training tasks) is used only for checkpoint selection and tuning.

Benchmark	Expert source	Nature of expert	Train	Val.	Test
StableToolBench	ToolBench traces (Qin et al., 2024) (solvable, finished)	model-generated	3,786 traj.	421	765 solvable queries (6 subsets)
τ-bench retail	ground-truth actions of the training tasks, replayed (below)	ground truth + simulated user	450 tasks	50	115 tasks, GPT-4o user
τ-bench retail (alt.)	APIGen-MT-5k (Prabhakar et al., 2025), retail domain	model-generated, verified	retail subset	50	as above
τ-bench airline	none (zero-shot transfer)	–	–	–	50 tasks, GPT-4o user
Mind2Web	released action sequences	human demonstrations	908 tasks	101	cross-task / -website / -domain
ALFWorld	ETO expert trajectories (Song et al., 2024)	rule-based expert	2,989 traj.	332	134 unseen games
WebShop	ETO expert trajectories (Song et al., 2024)	human and rule-based	1,744 traj.	194	200 instructions
BFCL-v3 (zero-shot)	ToolACE corpus (Liu et al., 2025), training only	model-generated	–	–	Non-Live / Live

τ-bench expert dialogues. The τ-bench training split provides, for each task, the user instruction and the ground-truth sequence of database actions, but no dialogue. We construct expert dialogues by running an agent that executes exactly these ground-truth actions in the environment, in order, against the Qwen3.5-9B user simulator, with GPT-4o writing the natural-language turns (greetings, confirmations, and questions for missing information) around them. We keep an episode only if the environment's final database state matches the task's ground truth; 1,520 of 1,980 episodes pass (several per task), costing 3.1 GPU-hours plus API calls. This interaction happens once, before training, and is shared by all methods. Because the same simulator is used here and in training, all τ-bench test numbers use a different user simulator (GPT-4o, as in the original benchmark); Table 13 also reports the Qwen3.5-9B user, and Table 14 replaces our expert dialogues by the public retail-domain trajectories of APIGen-MT-5k, after removing those whose user instruction matches a test task.

Actions and decision equivalence. The expert action at a step is a database call or a `respond` turn (τ-bench), a tool call (StableToolBench, ToolACE), a `CLICK`/`TYPE`/`SELECT` on a target element (Mind2Web, with the standard candidate-generation stage of the benchmark), or a text command (ALFWorld, WebShop). Two tool calls are equivalent if they name the same tool and their arguments agree after normalization (whitespace, case, number formats, argument order, defaults filled in); two web actions if they have the same operation, target element, and normalized value; two commands if they match after normalization. Two `respond` actions are equivalent only if their normalized strings are identical, which is conservative (Appendix H).

B Implementation Details and Tuning

Losses and optimization. The SFT loss (warm-up and anchor) is averaged over the action tokens of each example and over the batch; the pair log-probabilities in Eq. (1) are summed over action tokens, and the loss is averaged over pairs. All models are fully fine-tuned in bf16 with AdamW (weight decay 0) and a constant learning rate after a 5% linear warm-up. On ALFWorld and WebShop the rendered action includes the ReAct thought, and decisions are compared on the command only.

Decoding and evaluation. Evaluation decodes greedily. On τ-bench the user simulator samples at temperature 1.0, so the four trials per task differ; pass^1 averages them. SoPR uses the official StableToolBench evaluator with GPT-4o as the judge and the benchmark's cached tool responses.

Baselines. *RFT*: four samples per training state at temperature 1.0; samples decision-equivalent to the expert are added to the SFT data. *ETO*: failed trajectories from exploration with the warm-up student (four per task), paired with the expert trajectory of the same task, trained with trajectory-level DPO plus our SFT anchor for two rounds. On static Mind2Web, where nothing can be executed, the failed trajectory is the expert trajectory with the warm-up student's step-wise predictions substituted at the steps where they disagree with the expert, so no action is executed; the trajectory-pair controls of Table 4c use the same construction there. *IPR*: step rewards from Monte-Carlo rollouts of the warm-up student from each expert prefix, with the step-level DPO objective of Xiong et al. (2024). *PivotRL*: one-step samples at every expert state from the frozen warm-up student for profiling ($K = 8$ samples), keeping turns with nonzero reward variance and mean reward below λ_{diff}; the verifier is our decision-equivalence check, the objective group-relative with a KL penalty to the warm-up student, as in Yi et al. (2026). *GRPO*: G rollouts per task with the Qwen3.5-9B user on τ-bench, the environment on ALFWorld and WebShop, and the StableToolBench cached tools with the SoPR evaluator's judgment as the reward; a KL penalty to the warm-up student.

Controls in Table 4. All controls use the same PPA views, anchor weight, number of updates, and checkpoint selection as Agentic-DPO, and each control's own weight is tuned over four values on the τ-bench validation split. *Unlikelihood*: $-w \sum_t \log(1 - \pi_\theta(a_t^- \mid a_{<t}^-, s))$ over the tokens of a^- from the first token where it departs from a^+, plus the anchor. *KTO*: the expert action as desirable and the negative as undesirable example, with the default loss-aversion weights and β tuned. *Group-relative RL on the same samples*: the $K = 4$ mined candidates of a state together with the expert action's equivalence reward form a group; each sample's advantage is its reward minus the group mean, divided by the group standard deviation, and the update is the clipped policy gradient with the anchor. *IPO*: τ tuned. *CPO*: the preference term without a reference model plus the anchor. *Hinge*: $\max\big(0, \gamma - (\log \pi_\theta(a^+) - \log \pi_\theta(a^-))/\max(|a^+|, |a^-|)\big)$. *SimPO*: β and γ tuned. *Linear push-down*: $\lambda \mathcal{L}_{\text{SFT}} + c \log \pi_\theta(a^- \mid s)$ with $c = \beta_{\text{eff}}/2$ and no reference model; it approximates the first-order expansion of Eq. (1), which also contains a small token-summed pull-up $-\frac{\beta_{\text{eff}}}{2} \log \pi_\theta(a^+ \mid s)$ that the control drops. *Step-level SPIN*: one sample per state from the current student without the decision filter (pairs whose negative equals the expert are dropped). *Off-policy negatives*: the most probable decision-different action among four samples of the untrained base model; rule-based negatives replace the expert's tool by another tool available at the state or perturb one argument. *Trajectory pairs with a random failure*: ETO's pairs (a failed exploration against the expert trajectory of the same task), trained with Agentic-DPO's β_{eff}, anchor, learning rate, and number of updates; the ETO baseline uses its own tuned configuration. *Trajectory pairs with the most probable failure*: among four failed explorations per task, the one with the highest mean per-step log-likelihood under the student, with the same settings. *Neutral views*: the recovery-context insertion with a successful, irrelevant observation instead of an error. The controls in blocks (a) and (b) of Table 4 train on the states mined by Agentic-DPO; Table 25 re-runs them with their own mining.

Tuning. Every method receives twelve configurations per benchmark, evaluated on the validation split with Qwen3.5-9B (Qwen3.5-2B for open-loop StableToolBench), and the selected configuration is reused for the other backbones (Table 6). Agentic-DPO's defaults come from a pilot on StableToolBench validation with Qwen3.5-2B (16 configurations, 58.0 GPU-hours) run before this study; its grid is centred on them, and the search selected them on every benchmark. Table 7 varies one hyperparameter at a time around them on validation, and Table 8 extends the β sweep toward zero.

Table 6: Tuning protocol. Twelve validation configurations per method and benchmark; tuning GPU-hours are for τ-retail with Qwen3.5-9B, approximated as twelve times the cost of the selected configuration, and are in addition to the final runs; the selected configuration is shown for τ-retail.

Method	Searched	Configs	Tuning GPU-h	Selected
SFT	lr $\{1,2,5\}\times10^{-5}$ × epochs $\{2,3\}$ × warm-up $\{0,3\%\}$	12	71.0	lr 2×10^{-5}, 3 epochs
PPA+SFT	as SFT	12	76.0	lr 2×10^{-5}, 3 epochs
RFT	lr $\{1,2,5\}\times10^{-5}$ × samples/state $\{4,8\}$ × temp. $\{0.7,1.0\}$	12	192.0	lr 1×10^{-5}, 4 samples/state
DFT	lr $\{1,2,5\}\times10^{-5}$ × updates $\{750,1500\}$ × clip $\{0,1\}$	12	106.0	lr 1×10^{-5}
ETO	β $\{0.05,0.1,0.2\}$ × lr $\{1,5\}\times10^{-6}$ × rounds $\{1,2\}$	12	283.0	$\beta=0.1$, $\lambda=0.5$, 2 rounds
IPR	rollouts/step $\{3,5\}$ × β $\{0.1,0.3\}$ × lr $\{1,2,5\}\times10^{-6}$	12	643.0	$\beta=0.1$, 5 MC rollouts/step
PivotRL	G $\{4,8\}$ × λ_{diff} $\{0.5,0.75\}$ × lr $\{0.5,1,2\}\times10^{-6}$	12	329.0	$G=8$, $\lambda_{\text{diff}}=0.75$, KL 0.01
GRPO (G=8)	lr $\{0.5,1,2\}\times10^{-6}$ × KL $\{0.01,0.04\}$ × G $\{4,8\}$	12	1346.0	$G=8$, lr 1×10^{-6}, KL 0.01
Agentic-DPO	β $\{0.004,0.008,0.016\}$ × λ $\{0.25,0.5\}$ × K $\{4,8\}$	12	200.0	pilot defaults ($\beta=0.008$, $\lambda=0.5$, $K=4$)

Table 7: Validation accuracy (STB canonical, Qwen3.5-2B) and success (τ-retail with the Qwen3.5-9B user, Qwen3.5-9B). *: default.

Setting	STB (val)	τ-retail (val)
$\beta=0.004$	90.8	41.3
$\beta=0.008^*$	91.2	41.9
$\beta=0.016$	91.0	41.6
$\beta=0.032$	89.9	40.6
$\lambda=0.25$	90.9	41.3
$\lambda=0.5^*$	91.2	41.9
$\lambda=1$	91.0	41.0
$\alpha=0$	90.3	40.5
$\alpha=0.5^*$	91.2	41.9
$\alpha=1$	90.9	42.0
temperature $=0.7$	91.0	41.5
temperature $=1^*$	91.2	41.9
temperature $=1.3$	90.6	41.7
$K=1$	87.3	31.4
$K=2$	90.6	40.7
$K=4^*$	91.2	41.9
$K=8$	91.0	41.6
learning rate $=1e-5$	90.6	41.1
learning rate $=2e-5^*$	91.2	41.9
learning rate $=5e-5$	89.8	40.2
$M=75$	90.7	41.2
$M=150^*$	91.2	41.9
$M=300$	90.9	41.5

Table 8: β toward zero on τ-retail (Qwen3.5-9B): validation success and test success (ten seeds).

β	Validation	Test
$\beta=0.0001$	31.5	$30.4_{\pm2.0}$
$\beta=0.0005$	37.2	$36.1_{\pm2.0}$
$\beta=0.001$	39.2	$38.3_{\pm1.7}$
$\beta=0.002$	40.6	$39.9_{\pm2.1}$
$\beta=0.004$	41.3	$40.6_{\pm1.6}$
$\beta=0.008^*$	41.9	$41.0_{\pm2.1}$
$\beta=0.016$	41.6	$40.8_{\pm1.4}$
$\beta=0.032$	40.6	$39.2_{\pm1.6}$

Checkpoint selection. Over the 160 selections behind Table 9 (four backbones, four metrics, ten seeds; for SoPR the round is selected by the solvable pass rate on the validation queries), the validation split picked round 1: 0, round 2: 22, round 3: 41, round 4: 50, round 5: 47. Table 9 compares the selected round with always taking the last one; the difference is at most 1.8 points.

Table 9: Validation-selected refresh round (as reported) vs. always the last round ($R=5$), for Agentic-DPO (ten seeds).

	STB canonical		STB perturbed		τ-retail		Mind2Web	
Backbone	selected	last	selected	last	selected	last	selected	last
Qwen3.5-2B	$90.3_{\pm1.7}$	$90.0_{\pm1.9}$	$85.0_{\pm2.0}$	$84.1_{\pm2.2}$	$33.3_{\pm2.6}$	$34.0_{\pm2.2}$	$51.0_{\pm2.2}$	$49.9_{\pm2.4}$
Qwen3.5-4B	$91.3_{\pm1.6}$	$90.5_{\pm1.5}$	$88.4_{\pm1.9}$	$88.2_{\pm2.3}$	$37.2_{\pm1.9}$	$35.4_{\pm1.6}$	$55.3_{\pm2.6}$	$54.9_{\pm2.9}$
Qwen3.5-9B	$93.6_{\pm1.2}$	$93.5_{\pm1.1}$	$91.5_{\pm1.3}$	$91.0_{\pm1.6}$	$41.0_{\pm2.1}$	$40.3_{\pm2.2}$	$63.6_{\pm1.8}$	$62.0_{\pm1.9}$
Llama-3.1-8B	$92.5_{\pm1.5}$	$91.6_{\pm1.8}$	$88.7_{\pm1.7}$	$88.4_{\pm1.5}$	$37.7_{\pm2.2}$	$36.0_{\pm1.4}$	$59.6_{\pm2.0}$	$59.0_{\pm2.3}$

C Additional Results

With Qwen3-32B (five seeds), GRPO started from Agentic-DPO has the highest mean on all five tasks; its difference to GRPO ranges from 0.6 to 5.5 points.

The ranking is the same with the Qwen3.5-9B user (Agentic-DPO minus GRPO: -0.9 points), and with the APIGen-MT-5k experts Agentic-DPO improves over PPA+SFT by 14.4 points.

Table 10: All methods on all backbones (mean±std over ten seeds; validation-selected checkpoints). STB: StableToolBench next-action accuracy (canonical, perturbed); τ-retail and SoPR: closed loop; Mind2Web: step success. Bold and † as in Table 3. GRPO is trained end to end with the SoPR reward on StableToolBench and evaluated on both of its metrics.

Method	STB canonical	STB perturbed	τ-retail	STB SoPR	Mind2Web
Qwen3.5-2B					
SFT	$57.4_{\pm 3.0}$	$52.6_{\pm 2.9}$	$17.1_{\pm 2.2}$	$23.4_{\pm 0.9}$	$36.0_{\pm 2.1}$
PPA+SFT	$83.7_{\pm 2.3}$	$81.4_{\pm 2.6}$	$21.1_{\pm 2.5}$	$33.4_{\pm 1.1}$	$41.6_{\pm 2.3}$
RFT	$85.0_{\pm 2.1}$	$82.6_{\pm 2.4}$	$23.8_{\pm 1.9}$	$33.7_{\pm 0.9}$	$43.7_{\pm 2.2}$
DFT	$85.9_{\pm 3.1}$	$83.6_{\pm 3.3}$†	$27.2_{\pm 2.3}$	$33.2_{\pm 1.4}$	$49.5_{\pm 2.5}$†
ETO	$87.6_{\pm 1.7}$	**85.0**$_{\pm 1.9}$	$28.5_{\pm 2.1}$	$35.1_{\pm 1.9}$	$48.3_{\pm 2.2}$
IPR	$87.2_{\pm 1.9}$	$84.4_{\pm 2.1}$†	$29.7_{\pm 1.3}$	$36.5_{\pm 2.2}$	–
PivotRL	$88.4_{\pm 1.6}$	$84.9_{\pm 1.9}$†	$33.0_{\pm 2.2}$†	$37.3_{\pm 1.2}$	$48.2_{\pm 2.1}$
GRPO	$86.9_{\pm 2.1}$	$83.8_{\pm 2.5}$†	**34.5**$_{\pm 1.4}$	$37.7_{\pm 2.0}$	–
Agentic-DPO	**90.3**$_{\pm 1.7}$	**85.0**$_{\pm 2.0}$	$33.3_{\pm 2.6}$†	**41.3**$_{\pm 1.8}$	**51.0**$_{\pm 2.2}$
Qwen3.5-4B					
SFT	$66.0_{\pm 2.6}$	$60.8_{\pm 2.8}$	$22.1_{\pm 2.2}$	$28.3_{\pm 1.0}$	$39.0_{\pm 2.2}$
PPA+SFT	$83.0_{\pm 2.0}$	$78.1_{\pm 3.2}$	$27.0_{\pm 2.0}$	$37.4_{\pm 1.7}$	$46.6_{\pm 2.6}$
RFT	$83.9_{\pm 1.9}$	$81.9_{\pm 2.7}$	$29.1_{\pm 2.5}$	$38.8_{\pm 1.6}$	$47.1_{\pm 2.4}$
DFT	$86.7_{\pm 1.7}$	$83.6_{\pm 2.2}$	$29.0_{\pm 2.5}$	$39.1_{\pm 1.5}$	$49.4_{\pm 2.3}$
ETO	$89.3_{\pm 1.4}$	$86.6_{\pm 1.7}$	$33.1_{\pm 2.6}$	$41.1_{\pm 2.1}$	$52.2_{\pm 2.0}$
IPR	$89.6_{\pm 1.6}$	$86.8_{\pm 2.0}$†	$34.9_{\pm 2.0}$	$42.2_{\pm 1.8}$	–
PivotRL	$89.4_{\pm 1.5}$	$86.6_{\pm 1.8}$	$35.4_{\pm 1.7}$	$42.9_{\pm 1.9}$	$52.9_{\pm 2.1}$
GRPO	$87.0_{\pm 1.9}$	$85.4_{\pm 2.6}$	**39.1**$_{\pm 2.2}$	$42.0_{\pm 1.4}$	–
Agentic-DPO	**91.3**$_{\pm 1.6}$	**88.4**$_{\pm 1.9}$	$37.2_{\pm 1.9}$†	**45.4**$_{\pm 1.7}$	**55.3**$_{\pm 2.6}$
Qwen3.5-9B					
SFT	$78.9_{\pm 2.3}$	$74.6_{\pm 2.4}$	$22.4_{\pm 1.7}$	$38.0_{\pm 1.4}$	$46.0_{\pm 2.2}$
PPA+SFT	$92.4_{\pm 1.5}$†	$90.3_{\pm 1.2}$	$25.8_{\pm 2.1}$	$46.3_{\pm 1.1}$	$52.7_{\pm 2.0}$
RFT	$92.6_{\pm 1.4}$†	$90.5_{\pm 1.3}$†	$28.4_{\pm 1.8}$	$47.2_{\pm 1.5}$	$54.2_{\pm 1.9}$
DFT	$93.1_{\pm 1.6}$†	$91.4_{\pm 1.8}$†	$33.1_{\pm 2.1}$	$48.0_{\pm 1.9}$	$56.0_{\pm 1.8}$
ETO	$93.0_{\pm 1.5}$†	$91.2_{\pm 1.6}$†	$31.6_{\pm 2.2}$	$49.4_{\pm 2.5}$	$59.0_{\pm 1.9}$
IPR	$93.0_{\pm 1.4}$†	$90.9_{\pm 1.5}$†	$36.0_{\pm 2.1}$	$49.9_{\pm 2.0}$	–
PivotRL	$93.2_{\pm 1.3}$†	$91.1_{\pm 1.4}$†	$38.9_{\pm 2.2}$	$51.0_{\pm 1.6}$	$59.8_{\pm 1.8}$
GRPO	$92.9_{\pm 1.4}$†	$90.6_{\pm 1.6}$†	**42.1**$_{\pm 1.7}$	$51.5_{\pm 1.9}$†	–
Agentic-DPO	**93.6**$_{\pm 1.2}$	**91.5**$_{\pm 1.3}$	$41.0_{\pm 2.1}$†	**53.2**$_{\pm 1.1}$	**63.6**$_{\pm 1.8}$
Llama-3.1-8B					
SFT	$74.2_{\pm 2.5}$	$69.6_{\pm 2.7}$	$19.5_{\pm 1.5}$	$34.0_{\pm 1.7}$	$42.4_{\pm 2.2}$
PPA+SFT	$89.2_{\pm 1.8}$	$87.6_{\pm 2.0}$†	$23.6_{\pm 1.7}$	$42.0_{\pm 1.1}$	$49.0_{\pm 2.3}$
RFT	$89.7_{\pm 1.7}$	$87.5_{\pm 1.9}$	$24.5_{\pm 1.8}$	$43.6_{\pm 1.2}$	$50.4_{\pm 2.2}$
DFT	$90.9_{\pm 1.9}$†	$88.8_{\pm 2.1}$†	$27.6_{\pm 3.2}$	$44.1_{\pm 1.8}$	$52.2_{\pm 2.1}$
ETO	$92.2_{\pm 1.6}$†	**89.3**$_{\pm 1.8}$	$27.8_{\pm 2.0}$	$47.5_{\pm 2.1}$	$54.5_{\pm 2.3}$
IPR	$90.0_{\pm 1.7}$	$87.9_{\pm 1.9}$†	$33.1_{\pm 2.2}$	$48.2_{\pm 2.7}$†	–
PivotRL	$90.7_{\pm 1.5}$	$88.8_{\pm 1.8}$†	$33.8_{\pm 2.2}$	$50.1_{\pm 1.1}$†	$56.0_{\pm 2.1}$
GRPO	$89.5_{\pm 1.9}$	$87.5_{\pm 2.3}$†	**37.9**$_{\pm 1.5}$	$47.4_{\pm 2.0}$	–
Agentic-DPO	**92.5**$_{\pm 1.5}$	$88.7_{\pm 1.7}$†	$37.7_{\pm 2.2}$†	**50.5**$_{\pm 1.6}$	**59.6**$_{\pm 2.0}$

Table 11: Closed-loop suite with Llama-3.1-8B (ten seeds; as Table 2).

Method	τ-retail	τ-airline	ALFWorld	WebShop	STB SoPR
PPA+SFT	$23.6_{\pm 1.7}$	$13.2_{\pm 1.8}$	$62.6_{\pm 3.6}$	$57.4_{\pm 1.2}$	$42.0_{\pm 1.1}$
RFT	$24.5_{\pm 1.8}$	$14.4_{\pm 2.4}$	$64.8_{\pm 3.3}$	$60.2_{\pm 1.7}$	$43.6_{\pm 1.2}$
DFT	$27.6_{\pm 3.2}$	$15.0_{\pm 3.0}$	$65.1_{\pm 2.8}$	$61.5_{\pm 2.4}$	$44.1_{\pm 1.8}$
ETO	$27.8_{\pm 2.0}$	$17.1_{\pm 2.3}$	$69.2_{\pm 2.8}$	$65.6_{\pm 2.1}$†	$47.5_{\pm 2.1}$
IPR	$33.1_{\pm 2.2}$	$17.9_{\pm 3.0}$†	$73.7_{\pm 2.9}$†	$64.1_{\pm 1.7}$†	$48.2_{\pm 2.7}$†
PivotRL	$33.8_{\pm 2.2}$	$19.9_{\pm 2.7}$†	$72.2_{\pm 3.0}$	$63.6_{\pm 2.9}$†	$50.1_{\pm 1.1}$†
GRPO	**37.9**$_{\pm 1.5}$	$20.2_{\pm 2.4}$†	**77.0**$_{\pm 2.4}$	**66.1**$_{\pm 2.6}$	$47.4_{\pm 2.0}$
Agentic-DPO	$37.7_{\pm 2.2}$†	**22.4**$_{\pm 3.5}$	$74.7_{\pm 2.2}$†	$65.2_{\pm 1.6}$†	**50.5**$_{\pm 1.6}$
Agentic-DPO → GRPO	$42.9_{\pm 2.0}$	$24.1_{\pm 2.4}$	$79.9_{\pm 2.3}$	$68.0_{\pm 1.6}$	$52.3_{\pm 1.4}$
Agentic-DPO − GRPO	−0.2 [−3.0, +2.5]	+2.2 [−2.3, +6.8]	−2.3 [−6.1, +1.5]	−0.9 [−3.3, +1.5]	+3.1 [+0.8, +5.4]
Agentic-DPO − PivotRL	+3.9 [+0.8, +7.1]	+2.5 [−1.6, +6.4]	+2.5 [−1.7, +6.7]	+1.6 [−1.0, +4.3]	+0.4 [−1.7, +2.5]
Ag.-DPO→GRPO − GRPO	+5.0 [+1.8, +8.0]	+3.9 [−0.3, +8.3]	+2.9 [−0.9, +6.6]	+1.9 [−0.5, +4.3]	+4.9 [+2.5, +7.1]

Table 12: Closed-loop suite with Qwen3-32B (five seeds; 8 H100 GPUs).

Method	τ-retail	τ-airline	ALFWorld	WebShop	STB SoPR
PPA+SFT	$43.9_{\pm 2.7}$	$27.0_{\pm 1.9}$	$80.8_{\pm 2.7}$	$67.4_{\pm 1.7}$	$55.4_{\pm 2.3}$
PivotRL	$50.9_{\pm 1.3}$	$29.6_{\pm 3.3}$	$85.4_{\pm 3.0}$	$70.3_{\pm 1.3}$	$58.5_{\pm 2.1}$
GRPO	$53.9_{\pm 2.4}$	$32.3_{\pm 3.7}$	$86.4_{\pm 1.7}$	$73.3_{\pm 1.2}$	$58.4_{\pm 0.9}$
Agentic-DPO	$51.5_{\pm 2.5}$	$36.0_{\pm 3.5}$	$86.1_{\pm 1.3}$	$71.8_{\pm 1.6}$	$60.1_{\pm 1.0}$
Agentic-DPO → GRPO	$56.5_{\pm 1.9}$	$37.8_{\pm 2.1}$	$89.7_{\pm 1.8}$	$73.9_{\pm 1.7}$	$61.4_{\pm 1.5}$

Element equivalence on Mind2Web. Decision equivalence is least obvious for web elements: 31 of the 49 valid alternatives in the Mind2Web audit click an ancestor or descendant of the expert's element. Treating such targets as equivalent in the filter changes step success by +0.3 points ($p = 0.71$).

Table 13: τ-retail with Qwen3.5-9B (ten seeds): pass^k from the four trials per task (GPT-4o user), and success with the Qwen3.5-9B user of training and validation.

Method	pass^1	pass^2	pass^4	Qwen user
PPA+SFT	$25.8_{\pm 2.1}$	14.1	6.8	$24.3_{\pm 1.2}$
ETO	$31.6_{\pm 2.2}$	18.5	10.2	$30.7_{\pm 2.8}$
PivotRL	$38.9_{\pm 2.2}$	24.9	15.0	$38.2_{\pm 2.8}$
GRPO	$42.1_{\pm 1.7}$	28.1	17.4	$41.2_{\pm 2.4}$
Agentic-DPO	$41.0_{\pm 2.1}$	27.2	16.2	$40.3_{\pm 2.8}$
Agentic-DPO → GRPO	$46.2_{\pm 2.6}$	32.5	22.1	$45.6_{\pm 1.8}$

Table 14: τ-retail (9B, GPT-4o user) with our expert dialogues vs. the public APIGen-MT-5k retail trajectories as experts.

Method	Ours	APIGen-MT
PPA+SFT	$25.8_{\pm 2.1}$	$27.9_{\pm 2.1}$
ETO	$31.6_{\pm 2.2}$	$33.0_{\pm 2.0}$
Agentic-DPO	$41.0_{\pm 2.1}$	$42.3_{\pm 1.5}$

Table 15: Mind2Web with Qwen3.5-9B per split (cross-task / cross-website / cross-domain; means over ten seeds). Step success averaged over the splits is the Mind2Web column of Table 3.

Method	Element accuracy	Operation F1	Step success	Task success
PPA+SFT	62.5 / 56.4 / 54.8	84.2 / 82.6 / 83.1	57.4 / 51.3 / 49.4	5.1 / 3.4 / 2.8
ETO	68.1 / 62.7 / 60.9	86.9 / 85.2 / 85.8	63.4 / 57.9 / 55.7	7.9 / 5.6 / 4.8
PivotRL	68.9 / 63.4 / 61.7	87.1 / 85.5 / 85.9	64.2 / 58.7 / 56.5	8.3 / 5.8 / 5.0
Agentic-DPO	72.3 / 66.8 / 65.4	88.1 / 86.4 / 86.9	67.9 / 62.1 / 60.8	9.8 / 6.9 / 6.1

Table 16: Zero-shot BFCL-v3 accuracy of Qwen3.5-9B trained on ToolACE (ten seeds).

Method	Non-Live	Live
Base model (single evaluation)	82.7	56.1
SFT	$81.7_{\pm 0.6}$	$51.8_{\pm 1.2}$
PPA+SFT	$82.9_{\pm 0.5}$	$55.8_{\pm 1.0}$
Agentic-DPO	$84.1_{\pm 0.5}$	$57.3_{\pm 0.9}$

Table 17: Errors on canonical StableToolBench (Qwen3.5-2B), as % of each method's errors.

Method	Format/parse	Wrong tool	Wrong arguments
SFT	61	22	17
PPA+SFT	9	48	43
Agentic-DPO	6	45	49

Plain SFT. Plain SFT is weak on canonical StableToolBench because of a format mismatch rather than undertraining: the test prompts use the benchmark's function-calling template, whereas the released training traces use ToolBench's ReAct rendering, so SFT trained on the released rendering alone makes mostly format and parse errors (Table 17). PPA's action-rendering views include the test rendering among others. SFT and PPA+SFT use the same number of updates (PPA's renderings add some tokens, hence the slightly higher cost in Table 34), and every method starts from PPA+SFT, which is therefore the reference throughout.

Table 18: Untrained base models ($\pm$: three sampled evaluations; τ-retail with the Qwen3.5-9B user). These use sampled decoding and, on τ-bench, the training-time user simulator, unlike the greedy decoding and GPT-4o test user of the trained models, so they are a reference point rather than a baseline.

Backbone	STB canonical	STB perturbed	τ-retail	Mind2Web
Qwen3.5-2B	$45.2_{\pm 2.1}$	$42.1_{\pm 2.3}$	$12.5_{\pm 1.5}$	$25.6_{\pm 1.8}$
Qwen3.5-4B	$50.5_{\pm 2.2}$	$48.2_{\pm 2.4}$	$14.5_{\pm 1.6}$	$28.4_{\pm 1.9}$
Qwen3.5-9B	$65.2_{\pm 2.3}$	$62.1_{\pm 2.5}$	$16.5_{\pm 1.8}$	$35.2_{\pm 2.1}$
Llama-3.1-8B	$58.4_{\pm 2.4}$	$54.9_{\pm 2.6}$	$14.2_{\pm 1.7}$	$31.8_{\pm 2.0}$

D Statistical Procedures and Tests

Closed-loop intervals. For a difference between two methods on a closed-loop benchmark we resample the test tasks with replacement (the same resample for both methods, since they are evaluated on the same tasks) and, independently for each method, its ten training seeds; the statistic is the difference of the mean scores over the resampled tasks and seeds. We report the 2.5th and 97.5th percentiles of 4,000 resamples, and two-sided p-values as twice the smaller tail mass at zero. Shares of a gain use the same resamples. This accounts for the finite test sets as well as for training randomness; seed-only tests would give narrower intervals.

Open-loop tests and families. Open-loop comparisons (next-action accuracy on 1,100 to 7,700 test steps) use two-sided Welch tests over seeds. The primary family, fixed before the experiments, is Agentic-DPO vs. PPA+SFT on τ-retail across the four backbones (Holm). The table marks (†)

compare each method with the best one in its cell without correction, which makes "not significantly different" conservative for Agentic-DPO's claims. For the mechanism study, Table 22 reports raw and Holm-adjusted p over all 62 comparisons. Table 23 decomposes the four pair constructions of Table 4 (trajectory vs. same state, random vs. most probable negative) into simple effects and their interaction.

Table 19: Per-seed τ-retail success (GPT-4o user) with Qwen3.5-9B; the mean and standard deviation are those of Table 2.

Method	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	Mean	Std
PPA+SFT	23.0	27.2	26.3	24.1	28.7	24.8	27.2	23.3	24.6	28.7	25.8	2.1
RFT	27.0	25.7	27.6	28.9	28.3	27.8	28.5	32.4	27.6	30.2	28.4	1.8
DFT	32.2	34.1	29.6	34.8	34.6	31.3	30.2	34.3	34.6	35.2	33.1	2.1
ETO	30.9	32.4	32.0	29.8	30.7	36.1	28.0	30.4	32.2	33.5	31.6	2.2
IPR	36.3	35.4	35.9	30.7	35.4	36.5	37.6	37.8	38.0	36.3	36.0	2.1
PivotRL	40.4	38.9	38.7	42.6	40.0	39.1	35.9	35.0	39.8	38.5	38.9	2.2
GRPO	45.0	43.5	43.3	40.9	43.3	40.9	40.2	40.0	41.5	42.4	42.1	1.7
Agentic-DPO	36.5	41.1	40.4	40.9	41.1	42.8	40.0	44.8	40.2	42.2	41.0	2.1
Agentic-DPO $\rightarrow$ GRPO	45.4	49.1	47.0	43.0	50.9	42.8	44.3	47.6	46.7	45.2	46.2	2.6

Table 20: Closed-loop suite, Qwen3.5-9B: Agentic-DPO minus each baseline, with 95% bootstrap intervals over tasks and seeds.

Agentic-DPO minus	τ-retail	τ-airline	ALFWorld	WebShop	STB SoPR
PPA+SFT	+15.2 [+11.8, +18.7]	+8.2 [+4.0, +12.7]	+10.1 [+6.0, +14.3]	+5.9 [+3.8, +8.1]	+6.9 [+4.9, +8.9]
RFT	+12.6 [+9.5, +15.7]	+7.0 [+3.0, +11.3]	+7.3 [+3.1, +11.6]	+4.4 [+2.1, +6.7]	+6.0 [+3.9, +8.1]
DFT	+7.9 [+4.9, +11.0]	+5.9 [+1.9, +10.0]	+7.8 [+3.9, +11.9]	+4.6 [+2.4, +6.8]	+5.2 [+3.1, +7.3]
ETO	+9.4 [+6.5, +12.4]	+5.1 [+0.9, +9.7]	+3.9 [+0.1, +7.7]	+1.5 [−0.6, +3.7]	+3.8 [+1.4, +6.2]
IPR	+5.0 [+1.9, +8.1]	+2.8 [−1.0, +6.6]	+0.8 [−2.6, +4.4]	+0.6 [−1.6, +2.9]	+3.3 [+1.1, +5.5]
PivotRL	+2.1 [−1.0, +5.2]	+2.0 [−1.8, +6.0]	+1.2 [−2.2, +4.9]	+0.5 [−1.7, +2.7]	+2.2 [+0.1, +4.3]
GRPO	−1.1 [−4.1, +1.8]	+1.4 [−2.4, +5.3]	−1.4 [−5.2, +2.3]	−0.9 [−3.5, +1.5]	+1.7 [−0.5, +3.9]

Table 21: Open-loop comparisons: Agentic-DPO minus PPA+SFT (Holm-adjusted over the four backbones) and minus the best other method (raw Welch p).

Metric	Backbone	vs. PPA+SFT	p_{Holm}	vs. best other	p
STB canonical	Qwen3.5-2B	+6.6	<0.001	+1.9 (PivotRL)	0.019
STB canonical	Qwen3.5-4B	+8.3	<0.001	+1.7 (IPR)	0.030
STB canonical	Qwen3.5-9B	+1.2	0.065	+0.4 (PivotRL)	0.487
STB canonical	Llama-3.1-8B	+3.3	<0.001	+0.3 (ETO)	0.671
STB perturbed	Qwen3.5-2B	+3.6	0.009	+0.0 (ETO)	1.000
STB perturbed	Qwen3.5-4B	+10.3	<0.001	+1.6 (IPR)	0.085
STB perturbed	Qwen3.5-9B	+1.2	0.091	+0.1 (DFT)	0.887
STB perturbed	Llama-3.1-8B	+1.1	0.199	−0.6 (ETO)	0.450
Mind2Web	Qwen3.5-2B	+9.4	<0.001	+1.5 (DFT)	0.170
Mind2Web	Qwen3.5-4B	+8.7	<0.001	+2.4 (PivotRL)	0.035
Mind2Web	Qwen3.5-9B	+10.9	<0.001	+3.8 (PivotRL)	<0.001
Mind2Web	Llama-3.1-8B	+10.6	<0.001	+3.6 (PivotRL)	<0.001

Table 22: Mechanism study: Agentic-DPO minus each variant in points, with (raw p; Holm-adjusted p over all 62 comparisons). τ-retail uses the bootstrap of this section, the other columns Welch's test.

Variant	STB canonical (2B)	τ-retail (9B)	Mind2Web (9B)
No negative: continued SFT on the mined states	+5.0 (<0.001; <0.001)	+12.0 (<0.001; <0.001)	+9.1 (<0.001; <0.001)
One on-policy sample, no decision filter (step-level SPIN)	+5.2 (<0.001; <0.001)	+11.6 (<0.001; <0.001)	+7.7 (<0.001; <0.001)
One filtered on-policy sample ($K = 1$)	+4.2 (<0.001; 0.002)	+10.6 (<0.001; <0.001)	+7.3 (<0.001; <0.001)
Sample from the untrained base model	+3.5 (<0.001; 0.015)	+9.7 (<0.001; <0.001)	+6.4 (<0.001; <0.001)
Rule-perturbed expert action	+4.3 (<0.001; 0.002)	+10.6 (<0.001; <0.001)	+7.2 (<0.001; <0.001)
Random valid sample among $K = 4$	+3.3 (<0.001; 0.017)	+4.7 (0.002; 0.046)	+3.3 (<0.001; 0.016)
One of the two most probable	+1.3 (0.088; 1.000)	+2.1 (0.155; 1.000)	+1.6 (0.048; 0.953)
Unlikelihood on the negative	+2.7 (0.003; 0.086)	+6.5 (<0.001; <0.001)	+5.0 (<0.001; <0.001)
KTO (unpaired)	+1.4 (0.077; 1.000)	+3.3 (0.035; 0.781)	+2.5 (0.004; 0.117)
Group-relative RL on the same K samples	+1.7 (0.038; 0.802)	+3.0 (0.060; 1.000)	+2.6 (0.004; 0.117)
Hinge	+1.4 (0.073; 1.000)	+1.9 (0.193; 1.000)	+2.2 (0.011; 0.265)
SimPO	+1.0 (0.182; 1.000)	+1.4 (0.350; 1.000)	+1.5 (0.066; 1.000)
IPO	+0.7 (0.341; 1.000)	+0.7 (0.645; 1.000)	+0.9 (0.244; 1.000)
CPO (no reference model)	+0.5 (0.509; 1.000)	+0.6 (0.689; 1.000)	+0.7 (0.385; 1.000)
Linear push-down: anchor $+ c \log \pi_\theta(a^-)$	+0.4 (0.597; 1.000)	+0.4 (0.800; 1.000)	+0.5 (0.546; 1.000)
Trajectory pairs, random failure	+2.5 (0.004; 0.117)	+8.8 (<0.001; <0.001)	+4.3 (<0.001; 0.003)
Trajectory pairs, most probable failure	+2.1 (0.011; 0.265)	+7.6 (<0.001; <0.001)	+3.7 (<0.001; 0.009)
No decision-equivalence filter	+5.4 (<0.001; <0.001)	+8.7 (<0.001; <0.001)	+6.0 (<0.001; <0.001)
Mine once ($R = 1$, same number of updates)	+2.3 (0.007; 0.182)	+5.5 (<0.001; <0.001)	+3.0 (0.001; 0.040)
Without the SFT anchor	+18.2 (<0.001; <0.001)	+16.7 (<0.001; <0.001)	+13.4 (<0.001; <0.001)
Without recovery-context views	+1.5 (0.073; 1.000)	+4.4 (0.008; 0.200)	–

Table 23: The 2×2 design of pairing unit (trajectory vs. same state) and selection (random vs. most probable): simple effects and the interaction (half the difference of differences, with p).

Benchmark	Same state minus trajectory: random	Same state minus trajectory: most probable	Most probable minus random: trajectory	Most probable minus random: same state	Interaction
StableToolBench, canonical (2B)	−0.8	+2.1	+0.4	+3.3	+1.4 (0.011)
StableToolBench, perturbed (2B)	−2.9	−0.3	+0.2	+2.8	+1.3 (0.042)
τ-retail (9B)	+4.1	+7.6	+1.2	+4.7	+1.8 (0.105)
Mind2Web (9B)	+1.0	+3.7	+0.6	+3.3	+1.4 (0.023)

E Mechanism Study: Full Table, Own Mining, Second Backbone

Table 24: All variants of the mechanism study (ten seeds; StableToolBench with Qwen3.5-2B, τ-retail and Mind2Web with Qwen3.5-9B).

Variant	STB canonical	STB perturbed	τ-retail	Mind2Web
PPA+SFT (starting point of every row)	$83.7_{\pm 2.3}$	$81.4_{\pm 2.6}$	$25.8_{\pm 2.1}$	$52.7_{\pm 2.0}$
Continued SFT on all states	$84.2_{\pm 2.1}$	$81.9_{\pm 2.4}$	$26.4_{\pm 2.0}$	$53.2_{\pm 1.9}$
No negative: continued SFT on the mined states	$85.3_{\pm 2.0}$	$82.8_{\pm 2.2}$	$29.0_{\pm 1.5}$	$54.5_{\pm 1.8}$
One on-policy sample, no decision filter (step-level SPIN)	$85.1_{\pm 2.0}$	$82.4_{\pm 2.3}$	$29.4_{\pm 1.3}$	$55.9_{\pm 1.9}$
One filtered on-policy sample ($K = 1$)	$86.1_{\pm 1.8}$	$82.9_{\pm 2.0}$	$30.4_{\pm 1.8}$	$56.3_{\pm 1.8}$
Sample from the untrained base model	$86.8_{\pm 1.9}$	$82.5_{\pm 2.2}$	$31.3_{\pm 1.9}$	$57.2_{\pm 1.9}$
Rule-perturbed expert action	$86.0_{\pm 1.9}$	$81.9_{\pm 2.2}$	$30.4_{\pm 2.3}$	$56.4_{\pm 1.9}$
Random valid sample among $K = 4$	$87.0_{\pm 1.8}$	$82.2_{\pm 2.1}$	$36.3_{\pm 1.7}$	$60.3_{\pm 1.6}$
One of the two most probable	$89.0_{\pm 1.5}$	$83.8_{\pm 1.9}$	$38.9_{\pm 2.1}$	$62.0_{\pm 1.5}$
Unlikelihood on the negative	$87.6_{\pm 1.8}$	$83.7_{\pm 2.1}$	$34.5_{\pm 1.9}$	$58.6_{\pm 1.7}$
KTO (unpaired)	$88.9_{\pm 1.6}$	$84.1_{\pm 1.9}$	$37.7_{\pm 2.4}$	$61.1_{\pm 1.6}$
Group-relative RL on the same K samples	$88.6_{\pm 1.7}$	$84.0_{\pm 2.0}$	$38.0_{\pm 1.9}$	$61.0_{\pm 1.7}$
Hinge	$88.9_{\pm 1.6}$	$84.2_{\pm 1.9}$	$39.1_{\pm 2.4}$	$61.4_{\pm 1.6}$
SimPO	$89.3_{\pm 1.5}$	$84.5_{\pm 1.8}$	$39.6_{\pm 2.3}$	$62.1_{\pm 1.6}$
IPO	$89.6_{\pm 1.5}$	$84.6_{\pm 1.7}$	$40.3_{\pm 2.0}$	$62.7_{\pm 1.5}$
CPO (no reference model)	$89.8_{\pm 1.6}$	$84.6_{\pm 1.9}$	$40.4_{\pm 2.3}$	$62.9_{\pm 1.7}$
Linear push-down: anchor $+ c \log \pi_\theta(a^-)$	$89.9_{\pm 1.6}$	$84.8_{\pm 1.9}$	$40.6_{\pm 2.3}$	$63.1_{\pm 1.8}$
Trajectory pairs, random failure	$87.8_{\pm 1.7}$	$85.1_{\pm 1.9}$	$32.2_{\pm 2.4}$	$59.3_{\pm 1.9}$
Trajectory pairs, most probable failure	$88.2_{\pm 1.6}$	$85.3_{\pm 1.8}$	$33.4_{\pm 2.5}$	$59.9_{\pm 1.8}$
No decision-equivalence filter	$84.9_{\pm 2.1}$	$80.6_{\pm 2.5}$	$32.3_{\pm 1.9}$	$57.6_{\pm 1.9}$
String-identity filter only	$86.4_{\pm 1.9}$	$82.1_{\pm 2.2}$	$34.2_{\pm 1.3}$	$58.8_{\pm 1.8}$
Mine once ($R = 1$, same number of updates)	$88.0_{\pm 1.7}$	$83.9_{\pm 1.9}$	$35.5_{\pm 2.0}$	$60.6_{\pm 1.7}$
Without the SFT anchor	$72.1_{\pm 4.0}$	$67.9_{\pm 4.3}$	$24.3_{\pm 2.0}$	$50.2_{\pm 3.1}$
Without action-rendering views	$88.1_{\pm 2.0}$	$77.9_{\pm 2.7}$	$39.2_{\pm 2.1}$	$61.0_{\pm 1.8}$
Without recovery-context views	$88.8_{\pm 1.8}$	$80.9_{\pm 2.4}$	$36.6_{\pm 1.9}$	–
Neutral views instead of recovery views	$88.9_{\pm 1.7}$	$81.3_{\pm 2.2}$	$37.1_{\pm 1.5}$	–
Most probable valid sample (Agentic-DPO)	$90.3_{\pm 1.7}$	$85.0_{\pm 2.0}$	$41.0_{\pm 2.1}$	$63.6_{\pm 1.8}$

Table 25: τ-retail (9B): controls trained on Agentic-DPO's mined states (as in Table 4) vs. re-mining with their own student.

Variant	Mined states	Own mining
Continued SFT on the mined states	$29.0_{\pm 1.5}$	$29.3_{\pm 2.0}$
Unlikelihood	$34.5_{\pm 1.9}$	$34.9_{\pm 1.8}$
KTO	$37.7_{\pm 2.4}$	$38.0_{\pm 1.3}$
Group-relative RL on the samples	$38.0_{\pm 1.9}$	$38.3_{\pm 2.6}$
IPO	$40.3_{\pm 2.0}$	$40.4_{\pm 1.3}$
Linear push-down	$40.6_{\pm 2.3}$	$40.8_{\pm 1.2}$

Table 26: τ-retail with Llama-3.1-8B (ten seeds) and share of Agentic-DPO's gain (%).

Variant	Success	Share
PPA+SFT	$23.6_{\pm 1.7}$	0
Continued SFT on the mined states	$26.9_{\pm 1.8}$	23
Step-level SPIN	$28.4_{\pm 2.7}$	34
$K = 1$	$28.4_{\pm 1.7}$	34
Unlikelihood	$32.0_{\pm 2.3}$	60
Random valid negative	$33.8_{\pm 1.7}$	72
Linear push-down	$37.4_{\pm 2.1}$	98
ETO	$27.8_{\pm 2.0}$	30
Agentic-DPO	$37.7_{\pm 2.2}$	100

Table 27: Further variants on τ-retail with Qwen3.5-9B (ten seeds) and difference to Agentic-DPO with 95% interval.

Variant	Success	Difference
Drop reply-versus-reply pairs	$40.2_{\pm 2.3}$	-0.8 [-3.6, +2.2]
Semantic equivalence for replies (LLM judge)	$40.7_{\pm 1.1}$	-0.3 [-3.2, +2.5]
Drop negatives an LLM judge flags as valid	$40.8_{\pm 2.2}$	-0.2 [-3.2, +2.7]
Loss on the first divergent token only	$38.2_{\pm 1.4}$	-2.8 [-5.8, +0.1]
Loss on the tokens after divergence only	$34.8_{\pm 2.4}$	-6.2 [-9.1, -3.3]
Train under the mining view ($\psi = \phi$)	$39.9_{\pm 1.6}$	-1.1 [-3.9, +1.7]
Length-normalized selection of the negative	$40.4_{\pm 1.8}$	-0.6 [-3.6, +2.2]

F Operating Regime and the Check of Proposition 2

On τ-bench with Qwen3.5-9B, over all pairs and rounds, $\beta_{\text{eff}}\Delta$ has median 0.021 and 95th percentile 0.083, so the sigmoid weight $\sigma(-\beta_{\text{eff}}\Delta)$ is at least 0.479 for 95% of pairs. Over training, $\log \pi_\theta(a^+)$ rises by 0.9 nats and $\log \pi_\theta(a^-)$ changes by -14.2 nats on average. The contrastive term accounts for 38% of the norm of the Adam update (median over steps), and the cosine between the updates from the two terms is 0.21.

Table 28 checks Proposition 2. For each rule we measure its coverage (the share of training states, among those where greedy decoding erred, at which it forms a pair), $\tilde{q}(a_1)$ (the share of its pairs whose negative is the greedy competitor), and the growth of the margin m after one round, from which we subtract the growth under anchor-only training to isolate the contrastive part; dividing by the coverage gives the contrastive gain per pair. The proposition predicts the per-pair ratio between two rules as $(1 + \tilde{q})/(1 + \tilde{q}_{\text{rand}})$, bounded by $2K/(K + 1) = 1.6$ for $K = 4$. For the three rules that select among four candidates, the measured ratios are close to the predictions. The single filtered sample is the exception: per pair it picks the greedy competitor about as often as the random rule ($\tilde{q} = 0.44$ vs. 0.43), yet reaches only 0.52 of the random rule's per-pair gain, against a first-order prediction of 1.01. The first-order model does not explain this shortfall. One plausible explanation involves the optimizer: by Proposition 1 an unselected sample's expected update is the SFT direction, which the anchor already follows, and Adam's per-coordinate normalization changes the update little when a gradient component is collinear with the dominant one; we have not isolated this effect. The prediction concerns one round from initialization; later rounds re-mine against a changed student.

Table 28: τ-retail (9B), first round: coverage (%), share of pairs whose negative is the greedy competitor ($\tilde{q}(a_1)$), predicted per-pair ratio of contrastive margin gains relative to the random rule, measured margin growth (nats; mean$\pm$std over ten seeds), its contrastive part (minus anchor-only), the contrastive part per pair (divided by coverage), and the measured per-pair ratio.

Negative rule	Coverage	$\tilde{q}(a_1)$	Predicted	Margin growth	Contrastive	Per pair	Measured
Most probable valid sample	99	0.77	1.24	$1.04_{\pm 0.16}$	0.77	0.78	1.26
One of the two most probable	99	0.47	1.03	$0.92_{\pm 0.15}$	0.65	0.66	1.07
Random valid sample	99	0.43	1.00	$0.88_{\pm 0.14}$	0.61	0.62	1.00
One filtered sample ($K = 1$)	69	0.44	1.01	$0.49_{\pm 0.11}$	0.22	0.32	0.52
Anchor only (continued SFT)	–	–	–	$0.27_{\pm 0.09}$	–	–	–

G WHERE AGENTIC-DPO AND RL HELP

Expert states and the student's own states. Table 29 gives the numbers behind Figure 2b for all methods, the error at the student's own states, and next-action accuracy along the test expert dialogues. The pass@8 of a held-out state is measured once, with the PPA+SFT student, by sampling eight actions at temperature 1.0. The bins and the greedy error use the tool-call states of held-out validation dialogues; next-action accuracy is measured on all steps of the test dialogues, with replies scored by the LLM judge of Appendix H, so the two columns are measured on different states and are not complementary. For the student's own states we sample 300 states from each method's test episodes (GPT-4o user) and ask a judge (GPT-4o, shown the state, the domain policy, and the action) whether the greedy action is a mistake; on 100 states labelled by two annotators the judge agrees with the human label with $\kappa = 0.74$.

Table 29: τ-retail (9B): greedy error (%) at held-out expert states by the PPA+SFT student's pass@8 (share of states: 16, 17, 25, 42%), over all expert states, and at the student's own states (judge; mean$\pm$std over ten seeds); next-action accuracy along the test expert dialogues.

	Expert states, by pass@8						
Method	**0/8**	**1–4/8**	**5–7/8**	**8/8**	**All**	**Own states**	**Next action**
PPA+SFT	96	64	18	1.5	31.4	$27.9_{\pm 1.9}$	70.8
DFT	80	38	9	1.2	22.0	–	75.9
ETO	84	41	10	1.3	23.5	–	74.6
PivotRL	90	20	4	1	19.2	$22.4_{\pm 1.7}$	79.2
GRPO	78	30	7	1.2	19.8	$16.1_{\pm 1.5}$	76.4
Agentic-DPO	52	14	3	0.8	11.8	$19.8_{\pm 1.6}$	83.1
Agentic-DPO $\rightarrow$ GRPO	47	11	2.5	0.7	10.3	$13.5_{\pm 1.4}$	84.0

Task horizon. Table 30 splits the τ-retail test tasks by the number of actions in their ground-truth solution. Agentic-DPO's gain over PPA+SFT is large on short and medium tasks and small on long ones; GRPO's advantage over Agentic-DPO is confined to the long tasks, where the student is most likely to leave the expert's state distribution. GRPO started from Agentic-DPO has the highest mean at every horizon, and on long tasks it is no better than GRPO.

Table 30: τ-retail (9B) success by task horizon (short: at most five ground-truth actions, 41 tasks; medium: six to nine, 46; long: ten or more, 28); means over ten seeds, and differences (computed from unrounded means) with 95% bootstrap intervals.

Method	**Short**	**Medium**	**Long**
PPA+SFT	46.1	18.8	7.6
PivotRL	63.8	32.2	13.3
GRPO	67.4	34.2	17.9
Agentic-DPO	68.4	34.5	11.7
Agentic-DPO $\rightarrow$ GRPO	73.7	38.8	18.3
Agentic-DPO $-$ PPA+SFT	+22.3 [+16.4, +28.1]	+15.7 [+11.0, +20.6]	+4.1 [+0.3, +8.7]
Agentic-DPO $-$ GRPO	+0.9 [$-$3.8, +5.8]	+0.3 [$-$4.2, +4.8]	$-$6.2 [$-$12.1, $-$1.3]
Ag.-DPO$\rightarrow$GRPO $-$ GRPO	+6.2 [+0.9, +11.6]	+4.6 [+0.4, +8.8]	+0.4 [$-$4.6, +5.2]

GRPO variants and initializations. Table 31 varies GRPO's group size, budget, and schedule, adds the expert SFT loss to its objective, and starts it from other methods.

Table 31: GRPO on τ-retail with Qwen3.5-9B (ten seeds): total GPU-hours (including the stage it starts from) and task success. Last three rows: difference in success between GRPO started from Agentic-DPO and from another method, with 95% bootstrap intervals over tasks and seeds.

Variant	GPU-h	Success
GRPO, $G = 4$	61.8	$40.3_{\pm 2.2}$
GRPO, $G = 8$ (Table 2)	112.2	$42.1_{\pm 1.7}$
GRPO, $G = 16$	219.5	$42.6_{\pm 2.6}$
GRPO, $G = 8$, twice as many updates	218.1	$42.7_{\pm 2.5}$
GRPO, $G = 8$, with the expert SFT loss mixed in	118.9	$42.4_{\pm 2.5}$
GRPO re-tuned for a 17 GPU-h budget	17.0	$33.9_{\pm 2.1}$
GRPO re-tuned for a 60 GPU-h budget	60.0	$39.6_{\pm 1.6}$
GRPO from PPA+SFT (Table 2)	112.2	$42.1_{\pm 1.7}$
GRPO from DFT	62.5	$43.0_{\pm 1.6}$
GRPO from ETO	77.3	$43.3_{\pm 2.6}$
GRPO from PivotRL	81.1	$44.0_{\pm 2.0}$
GRPO from Agentic-DPO	70.4	$46.2_{\pm 2.6}$
Agentic-DPO	16.7	$41.0_{\pm 2.1}$
GRPO from Agentic-DPO − from DFT		+3.2 [+0.2, +6.5]
GRPO from Agentic-DPO − from ETO		+2.9 [−0.6, +6.5]
GRPO from Agentic-DPO − from PivotRL		+2.2 [−1.0, +5.6]

H Negative Audit and Skipped States

Protocol. For each benchmark we sampled 100 mined negatives from rounds 1, 3, and 5 of two seeds, 200 random-rule negatives, and 200 expert actions as controls. Each item shows the state and *one* action; annotators do not know whether it is an expert action or a negative, and the three kinds are shuffled together. Two annotators independently label the action *wrong* (it would not lead to task completion or violates the domain policy), *valid* (it would lead to completion as well as the expert's action), or *undecidable*; disagreements are resolved by discussion. For a negative, "valid" means a false negative. On τ-bench, 37 of the 60 valid alternatives are rephrased `respond` turns. The LLM judge used for filtering in Table 27 (Llama-3.1-70B-Instruct, threshold chosen for recall on a held-out labelled set) receives the same view; against the human labels it has precision 0.58 and recall 0.82.

Table 32: Blind audit (% of n). Most probable negatives: wrong / valid alternative / undecidable and Cohen's κ; random negatives: wrong / valid; expert actions judged wrong.

		Most probable negatives				Random negatives		Expert
Benchmark	n	Wrong	Valid	Undec.	κ	Wrong	Valid	Wrong
StableToolBench (2B)	600	90.2	6.2	3.7	0.81	93.5	4.0	2.5
τ-retail (9B)	600	84.2	10.0	5.8	0.75	88.0	7.5	3.5
Mind2Web (9B)	600	87.0	8.2	4.8	0.79	90.5	6.0	2.0

Table 33: Skipped expert steps (%) by refresh round: steps at which no valid, decision-different candidate was sampled.

Benchmark	R1	R2	R3	R4	R5
StableToolBench (2B)	17.9	20.6	23.1	25.4	27.4
τ-retail (9B)	21.2	24.8	28.1	30.9	33.0
Mind2Web (9B)	12.4	14.3	16.5	18.2	19.8

I Compute Accounting

Costs are A6000 GPU-hours averaged over seeds and include every stage the reported number depends on: the PPA+SFT warm-up (shared by all methods that start from it), negative mining, profiling, exploration, rollouts or sampling, training, and validation for checkpoint selection (including user-simulator calls). Totals stop at the end of the schedule, not at the selected checkpoint. The τ-bench expert dialogues (Appendix A) are shared by all methods and not included. Per update, Agentic-DPO costs 1.6× the wall time of SFT, and GRPO with $G = 4$ costs 13.6×.

Table 34: Training cost (GPU-hours) on τ-retail with Qwen3.5-9B by stage, and total cost on ALFWorld, WebShop, and StableToolBench (SoPR training).

	τ-retail					ALFWorld	WebShop	STB
Method	**Warm-up**	**Sampling / rollouts**	**Training**	**Validation**	**Total**	**Total**	**Total**	**Total**
SFT	–	–	5.2	0.7	5.9	–	–	–
PPA+SFT	–	–	5.6	0.7	6.3	4.1	3.6	8.9
RFT	6.3	4.4 (sampling)	3.9	1.4	16.0	10.8	9.2	21.6
DFT	6.3	–	1.1	1.4	8.8	6.0	5.1	12.2
ETO	6.3	11.8 (exploration)	4.1	1.4	23.6	18.2	14.9	31.4
IPR	6.3	41.7 (MC rollouts)	4.2	1.4	53.6	40.6	33.9	77.0
PivotRL	6.3	4.9 (profiling)	14.8	1.4	27.4	19.7	16.4	38.1
GRPO (G=8)	6.3	104.5 (rollouts)	–	1.4	112.2	71.3	58.1	139.6
Agentic-DPO	6.3	4.4 (mining)	4.6	1.4	16.7	11.9	9.8	23.8
Agentic-DPO → GRPO	16.7	52.3 (GRPO stage)	–	1.4	70.4	47.5	39.3	91.2

J Stabilizers on a Second Backbone

Table 35: Stabilizer ablation on StableToolBench (ten seeds). The ablations remove a component from the preference stage; the warm-up is PPA+SFT in every row. Gap = canonical − perturbed.

Method	**Qwen3.5-2B**			**Gemma3-4B**		
	Canonical	**Perturbed**	**Gap**	**Canonical**	**Perturbed**	**Gap**
SFT	$57.4_{\pm 3.0}$	$52.6_{\pm 2.9}$	4.8	$66.8_{\pm 3.3}$	$51.8_{\pm 3.6}$	15.0
PPA+SFT	$83.7_{\pm 2.3}$	$81.4_{\pm 2.6}$	2.3	$73.2_{\pm 2.6}$	$69.8_{\pm 2.9}$	3.4
without the SFT anchor	$72.1_{\pm 4.0}$	$67.9_{\pm 4.3}$	4.2	$63.0_{\pm 4.5}$	$57.4_{\pm 4.9}$	5.6
without action-rendering views	$88.1_{\pm 2.0}$	$77.9_{\pm 2.7}$	10.2	$74.8_{\pm 2.3}$	$65.6_{\pm 3.0}$	9.2
without recovery-context views	$88.8_{\pm 1.8}$	$80.9_{\pm 2.4}$	7.9	$75.8_{\pm 2.1}$	$67.7_{\pm 2.7}$	8.1
without PPA (preference stage)	$87.6_{\pm 1.5}$	$76.3_{\pm 1.8}$	11.3	$74.2_{\pm 2.9}$	$64.3_{\pm 3.6}$	9.9
Full Agentic-DPO	$90.3_{\pm 1.7}$	$85.0_{\pm 2.0}$	5.3	$75.6_{\pm 1.7}$	$72.1_{\pm 2.0}$	3.5

The two PPA families are complementary: each alone recovers part of the perturbed accuracy lost without PPA, and together they recover more. Replacing the recovery-context views by neutral views (the same inserted call with a successful, irrelevant observation) gives 81.3 perturbed accuracy, against 80.9 without recovery views and 85.0 with them, so the error in the inserted history matters, not only the longer context. The full recipe's canonical–perturbed gap (5.3) remains larger than PPA+SFT's (2.3) with Qwen3.5-2B, whereas with Gemma3-4B the two gaps are similar: the stabilizers limit, but do not always remove, the format sensitivity that the contrast adds.

K Scaling in Candidates and Data, and Corrupted Experts

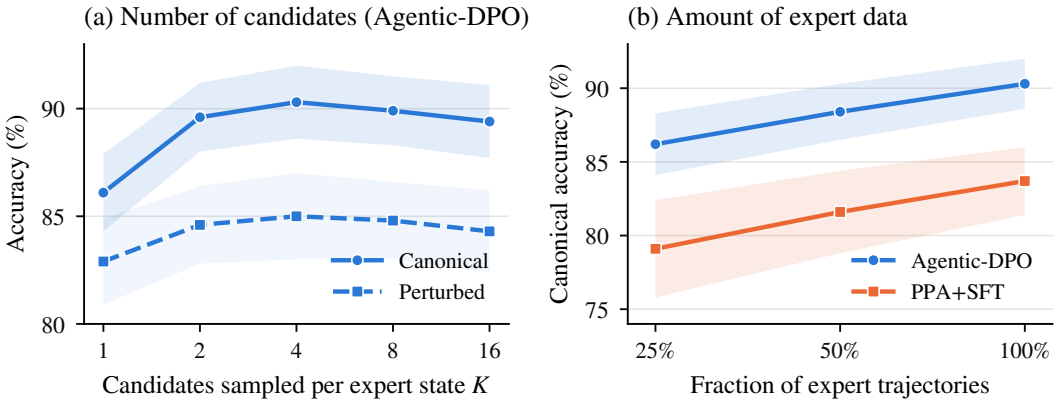

Figure 3: StableToolBench with Qwen3.5-2B; lines are means and bands ± 1 std over ten seeds. **(a)** Number of candidates K sampled per expert state. **(b)** Fraction of expert trajectories.

Two to four candidates per state suffice (Figure 3a); with a quarter of the expert trajectories, Agentic-DPO exceeds PPA+SFT trained on all of them on StableToolBench (by 2.5 points) and on τ-retail (by

8.2; Table 36). When a random valid action replaces 10% or 20% of the expert actions, Agentic-DPO degrades by 5.4 τ-retail points at 20%, PPA+SFT by 3.5 (Table 37).

Table 36: τ-retail (9B) with a fraction of the expert trajectories (ten seeds).

Method	25%	50%	100%
PPA+SFT	$20.3_{\pm 1.5}$	$23.0_{\pm 2.1}$	$25.8_{\pm 2.1}$
Agentic-DPO	$34.0_{\pm 1.5}$	$38.1_{\pm 1.4}$	$41.0_{\pm 2.1}$

Table 37: Corrupted expert actions: StableToolBench canonical (2B) and τ-retail (9B) with 0/10/20% of the expert actions replaced.

	StableToolBench			τ-retail		
Method	0%	10%	20%	0%	10%	20%
PPA+SFT	$83.7_{\pm 2.3}$	$80.9_{\pm 2.5}$	$77.2_{\pm 2.8}$	$25.8_{\pm 2.1}$	$24.2_{\pm 1.7}$	$22.3_{\pm 1.8}$
Agentic-DPO	$90.3_{\pm 1.7}$	$87.5_{\pm 1.9}$	$83.1_{\pm 2.1}$	$41.0_{\pm 2.1}$	$38.9_{\pm 1.5}$	$35.6_{\pm 1.7}$

L Proofs and Supporting Analysis

This appendix provides the full statements and proofs of the local results in Section 3.4: (i) at initialization, the contrastive gradient with unfiltered on-policy negatives is collinear with the SFT gradient in logit space (Proposition 1); (ii) a single anchored step produces a first-order decrease of the expert KL (Corollary 3); (iii) under the hard-negative rule used by Agentic-DPO, the contrastive gradient lowers only the selected competitors and remains aligned with the expert-KL descent direction (Proposition 4); and (iv) the most probable different action maximizes the first-order gain of the greedy-decision margin, by a factor of at most $2K/(K+1)$ over a random valid negative (Proposition 2).

L.1 Local analysis: setup and statements

This analysis uses a latent-action softmax abstraction. It treats each rendered action as the surface form of a latent action and writes the DPO coefficient as a fixed positive scalar β. In the implementation, this scalar is replaced by the length-scaled β_{eff} in Section 3.2; for a fixed action pair, this only changes the local gradient scale and not its direction, but in expectation over negatives of different lengths it also reweights negatives by length, which the statements below ignore.

Fix an expert decision state s with finite action set $\mathcal{A}(s)$. Let $z_\theta(s) \in \mathbb{R}^{|\mathcal{A}(s)|}$ denote the pre-softmax logits and $\pi_\theta(a \mid s) = \mathrm{softmax}(z_\theta(s))_a$. We write e_a for the standard basis vector of action a. Let π_{ref} be the frozen reference policy, and assume initialization satisfies

$$\pi_{\theta_0}(\cdot \mid s) \;=\; \pi_{\mathrm{ref}}(\cdot \mid s).$$

Let q_s denote the expert positive-action distribution at s (a Dirac $\delta_{a^*(s)}$ in the deterministic trajectory setting). For each state we sample a positive $a^+ \sim q_s$ and a negative $a^- \sim \pi_{\mathrm{ref}}(\cdot \mid s)$ independently; we assume $\pi_{\mathrm{ref}}(a \mid s) > 0$ on $\mathrm{supp}(q_s)$.

Define the reference-subtracted margin

$$\Delta_\theta(s, a^+, a^-) \;\triangleq\; \big[z_\theta(a^+ \,|\, s) - z_\theta(a^- \,|\, s)\big] - \big[z_{\mathrm{ref}}(a^+ \,|\, s) - z_{\mathrm{ref}}(a^- \,|\, s)\big],$$

the per-pair contrastive loss $\mathcal{L}_{\mathrm{con}}(\theta; s, a^+, a^-) = -\log \sigma(\beta\, \Delta_\theta(s, a^+, a^-))$, and the per-state expected losses

$$\bar{\mathcal{L}}_{\mathrm{con}}(\theta; s) \;\triangleq\; \mathbb{E}_{a^+ \sim q_s,\, a^- \sim \pi_{\mathrm{ref}}}\big[\mathcal{L}_{\mathrm{con}}\big], \qquad \bar{\mathcal{L}}_{\mathrm{SFT}}(\theta; s) \;\triangleq\; \mathbb{E}_{a^+ \sim q_s}\big[-\log \pi_\theta(a^+ \,|\, s)\big].$$

The joint anchored Agentic-DPO objective is $\bar{\mathcal{L}}_{\mathrm{train}}(\theta; s) \triangleq \bar{\mathcal{L}}_{\mathrm{con}}(\theta; s) + \lambda \bar{\mathcal{L}}_{\mathrm{SFT}}(\theta; s)$ with $\lambda > 0$, matching the anchored objective in Section 3.3 of the main paper.

Proof of Proposition 1. At initialization $\Delta_\theta = 0$, so for a single pair $\nabla_z\big[-\log \sigma(\beta \Delta_\theta)\big] = -\beta\, \sigma(-\beta \Delta_\theta) \nabla_z \Delta_\theta = \frac{\beta}{2}\,(e_{a^-} - e_{a^*})$, which is zero when $a^- = a^*$. Taking the expectation over $a^- \sim \tilde{q}$ gives $\frac{\beta}{2} \sum_{a \neq a^*} \tilde{q}(a)(e_a - e_{a^*}) = \frac{\beta}{2}(\tilde{q} - \delta_{a^*})$, because $\sum_{a \neq a^*} \tilde{q}(a) e_a = \tilde{q} - \tilde{q}(a^*) e_{a^*}$ and $\sum_{a \neq a^*} \tilde{q}(a) = 1 - \tilde{q}(a^*)$. With $\tilde{q} = \pi_{\mathrm{ref}}$ this is $\frac{\beta}{2}$ times the SFT gradient $\pi_{\mathrm{ref}} - \delta_{a^*}$; with $\tilde{q} = \pi_{\mathrm{ref}}(\cdot \mid a \neq a^*)$ it is $\frac{\beta}{2(1 - \pi_{\mathrm{ref}}(a^* | s))}$ times it. $\square$

Restatement of Proposition 1 (logit-space alignment at initialization). Under the setup above, at $\pi_{\theta_0} = \pi_{\mathrm{ref}}$ we have

$$\nabla_{z_{\theta_0}(s)} \bar{\mathcal{L}}_{\mathrm{con}}(\theta_0; s) \;=\; \tfrac{\beta}{2}\big(\pi_{\mathrm{ref}}(\cdot \,|\, s) - q_s\big), \qquad \nabla_{z_{\theta_0}(s)} \bar{\mathcal{L}}_{\mathrm{SFT}}(\theta_0; s) \;=\; \pi_{\mathrm{ref}}(\cdot \,|\, s) - q_s,$$

and therefore

$$\nabla_{z_{\theta_0}(s)} \bar{\mathcal{L}}_{\mathrm{train}}(\theta_0; s) \;=\; \Big(\lambda + \tfrac{\beta}{2}\Big)\big(\pi_{\mathrm{ref}}(\cdot \,|\, s) - q_s\big).$$

More generally, if the negative is drawn from any distribution $\tilde{q}(\cdot \mid s)$ with $\tilde{q}(a^*) = 0$, the same computation gives $\nabla_z \bar{\mathcal{L}}_{\mathrm{con}} = \frac{\beta}{2}(\tilde{q} - \delta_{a^*})$ at initialization. For a single sample filtered to differ from a^*, $\tilde{q} = \pi_{\mathrm{ref}}(\cdot \mid s, a \neq a^*)$, and $\tilde{q} - \delta_{a^*} = (\pi_{\mathrm{ref}} - \delta_{a^*})/(1 - \pi_{\mathrm{ref}}(a^* \mid s))$, which is the SFT gradient scaled by $1/(1 - \pi_{\mathrm{ref}}(a^* \mid s))$ (the second statement of Proposition 1). Selecting among $K > 1$ candidates yields a $\tilde{q}$ that is not proportional to π_{ref} off a^*; among single samples from the student, only then does the direction differ from SFT (off-policy negatives, e.g. from another model or from rules, also change it).

Corollary 3 (First-order decrease of expert KL)**.** *Let $V_s(z) \triangleq \mathrm{KL}(q_s \,\|\, \pi_z(\cdot \,|\, s))$ and $z' = z_{\theta_0}(s) - \eta \nabla_{z_{\theta_0}(s)} \bar{\mathcal{L}}_{\mathrm{train}}(\theta_0; s)$ for sufficiently small $\eta > 0$. Then*

$$V_s(z') \;=\; V_s(z_{\theta_0}(s)) - \eta\big(\lambda + \tfrac{\beta}{2}\big)\big\|\pi_{\mathrm{ref}}(\cdot \,|\, s) - q_s\big\|_2^2 + O(\eta^2).$$

In particular, $V_s(z') < V_s(z_{\theta_0}(s))$ whenever $\pi_{\mathrm{ref}}(\cdot \,|\, s) \neq q_s$.

The hard-negative variant of Corollary 3 (Proposition 4) is stated and proved in §L.3.

L.2 Proof of Corollary 3

Proof. $V_s(z) = \mathrm{KL}(q_s \,\|\, \pi_z)$ differs from the cross-entropy $-\sum_a q_s(a) \log \pi_z(a \mid s)$ by a z-independent constant, so by the softmax identity $\nabla_z V_s(z) = \pi_z - q_s$. At initialization this is $\pi_{\mathrm{ref}} - q_s$. By Proposition 1, $\nabla_{z_0} \bar{\mathcal{L}}_{\mathrm{train}} = (\lambda + \beta/2)\nabla_z V_s(z_0)$. A first-order Taylor expansion of V_s along $z' = z_0 - \eta \nabla_{z_0} \bar{\mathcal{L}}_{\mathrm{train}}$ gives

$$V_s(z') \;=\; V_s(z_0) - \eta\big(\lambda + \tfrac{\beta}{2}\big)\|\pi_{\mathrm{ref}} - q_s\|_2^2 + O(\eta^2),$$

which is strictly below $V_s(z_0)$ for sufficiently small η whenever $\pi_{\mathrm{ref}} \neq q_s$. □

L.3 Hard-Negative Extension of Corollary 3

We now extend the first-order KL-decrease statement to the hard-negative rule used in Agentic-DPO. The main text constructs negatives by sampling K candidates from the current student, removing invalid candidates and candidates equivalent to the expert action, and selecting the remaining candidate with the highest student log probability. For the local analysis, we study this rule at initialization and in the deterministic expert setting, which matches the expert trajectories used in our experiments.

Let $q_s = \delta_{a^*(s)}$ be a point mass on the expert action. Let $\tilde{q}_s^-$ denote the distribution of the selected hard negative after filtering, conditioned on the event that at least one valid non-expert candidate remains. By construction, $\tilde{q}_s^-(a^*(s)) = 0$.

Proposition 4 (Hard-negative gradient at initialization)**.** *Assume $q_s = \delta_{a^*(s)}$, $\tilde{q}_s^-(a^*(s)) = 0$, and $\pi_{\theta_0} = \pi_{\mathrm{ref}}$. Then*

$$\nabla_{z_{\theta_0}(s)} \bar{\mathcal{L}}_{\mathrm{con}}(\theta_0; s) = \tfrac{\beta}{2}\big(\tilde{q}_s^- - \delta_{a^*(s)}\big).$$

Moreover,

$$\Big\langle \nabla_z V_s(z_{\theta_0}(s)), \nabla_{z_{\theta_0}(s)} \bar{\mathcal{L}}_{\mathrm{con}}(\theta_0; s) \Big\rangle = \tfrac{\beta}{2}\Big(1 - \pi_{\mathrm{ref}}(a^*(s) \mid s) + \mathbb{E}_{a^- \sim \tilde{q}_s^-}[\pi_{\mathrm{ref}}(a^- \mid s)]\Big) \geq 0.$$

Therefore, for the anchored objective $\bar{\mathcal{L}}_{\mathrm{train}} = \bar{\mathcal{L}}_{\mathrm{con}} + \lambda \bar{\mathcal{L}}_{\mathrm{SFT}}$ with $\lambda > 0$, a sufficiently small gradient step produces a first-order decrease of $V_s(z) = \mathrm{KL}(\delta_{a^*(s)} \| \pi_z(\cdot \mid s))$ whenever $\pi_{\mathrm{ref}}(\cdot \mid s) \neq \delta_{a^*(s)}$.

Proof. At initialization $\pi_{\theta_0} = \pi_{\mathrm{ref}}$, the reference-subtracted margin is zero for every selected pair. Thus the same pointwise identity used in the proof of Proposition 1 applies:

$$\nabla_{z_{\theta_0}(s)} \mathcal{L}_{\mathrm{con}} = -\tfrac{\beta}{2}(e_{a^*} - e_{a^-}) = \tfrac{\beta}{2}(e_{a^-} - e_{a^*}).$$

Taking expectation over $a^- \sim \tilde{q}_s^-$ gives

$$\nabla_{z_{\theta_0}(s)} \bar{\mathcal{L}}_{\mathrm{con}}(\theta_0; s) = \tfrac{\beta}{2}\big(\tilde{q}_s^- - \delta_{a^*(s)}\big).$$

For $V_s(z) = \mathrm{KL}(\delta_{a^*(s)} \| \pi_z(\cdot \mid s))$, we have

$$\nabla_z V_s(z_{\theta_0}(s)) = \pi_{\mathrm{ref}}(\cdot \mid s) - \delta_{a^*(s)}.$$

Therefore

$$\begin{aligned}\Big\langle \pi_{\mathrm{ref}} - \delta_{a^*}, \tfrac{\beta}{2}(\tilde{q}_s^- - \delta_{a^*}) \Big\rangle &= \tfrac{\beta}{2}\left(\langle \pi_{\mathrm{ref}}, \tilde{q}_s^- \rangle - \pi_{\mathrm{ref}}(a^* \mid s) - \tilde{q}_s^-(a^*) + 1\right) \\ &= \tfrac{\beta}{2}\left(1 - \pi_{\mathrm{ref}}(a^* \mid s) + \mathbb{E}_{a^- \sim \tilde{q}_s^-}[\pi_{\mathrm{ref}}(a^- \mid s)]\right) \geq 0,\end{aligned}$$

where the equality uses $\tilde{q}_s^-(a^*) = 0$. The SFT part adds

$$\lambda \left\| \pi_{\mathrm{ref}}(\cdot \mid s) - \delta_{a^*(s)} \right\|_2^2 \geq 0$$

to the inner product with the KL gradient, and it is strictly positive whenever $\pi_{\mathrm{ref}}(\cdot \mid s) \neq \delta_{a^*(s)}$. A first-order Taylor expansion of V_s along the negative anchored-gradient direction then gives the claimed decrease. □

This proposition is intentionally stated for deterministic expert actions. For stochastic expert distributions, the same conclusion requires additional conditions on how the hard negative distribution relates to the support of q_s, which we do not need for the trajectory setting studied in this paper.

L.4 Proof of Proposition 2

We use the setup of Appendix L.1 with a deterministic expert $q_s = \delta_{a^*}$ and a constant β. Because a_1 is the unique maximizer of $z(a)$ over $a \neq a^*$, the margin $m(z) = z(a^*) - \max_{a \neq a^*} z(a)$ is differentiable in a neighborhood of $z_{\theta_0}(s)$ with gradient $e_{a^*} - e_{a_1}$. By Proposition 4, the expected contrastive gradient at initialization is $g = \frac{\beta}{2}(\tilde{q}_s^- - e_{a^*})$, with $\tilde{q}_s^-(a^*) = 0$. A step $z' = z - \eta g$ therefore changes the margin by

$$\langle e_{a^*} - e_{a_1}, -\eta g \rangle + O(\eta^2) = \tfrac{\eta\beta}{2}\big(1 - \tilde{q}_s^-(a^*) + \tilde{q}_s^-(a_1)\big) + O(\eta^2) = \tfrac{\eta\beta}{2}\big(1 + \tilde{q}_s^-(a_1)\big) + O(\eta^2).$$

For the ordering, fix the K candidates and let V be the set of valid candidates whose decision differs from a^*. If $a_1 \notin V$, none of the rules can select a_1. If $a_1 \in V$, then a_1 is the most probable element of V, so the most-probable rule selects it with probability 1; the top-2 rule selects it with probability $1/2$ if $|V| \geq 2$ and 1 otherwise; and the random rule selects it with probability $1/|V|$. Every rule forms a pair exactly when $V \neq \emptyset$, so taking expectations over the candidates conditioned on this event, as in the definition of $\tilde{q}_s^-$, gives $\tilde{q}_{\mathrm{mp}}^-(a_1) \geq \tilde{q}_{\mathrm{top2}}^-(a_1) \geq \tilde{q}_{\mathrm{rand}}^-(a_1)$, and the ordering follows. For the ratio, let $q = \Pr(a_1 \in V \mid V \neq \emptyset) = \tilde{q}_{\mathrm{mp}}^-(a_1)$. Since $|V| \leq K$, the random rule selects a_1 with probability at least $1/K$ whenever $a_1 \in V$, so $\tilde{q}_{\mathrm{rand}}^-(a_1) \geq q/K$ and

$$\frac{1 + \tilde{q}_{\mathrm{mp}}^-(a_1)}{1 + \tilde{q}_{\mathrm{rand}}^-(a_1)} \;\leq\; \frac{1+q}{1+q/K} \;\leq\; \frac{2K}{K+1},$$

because $(1+q)/(1+q/K)$ is increasing in $q \in [0,1]$. For $K = 4$ the bound is 1.6. □

L.5 Scale of the Two Terms in Practice

All main results use $\beta = 0.008$ with the length-scaled β_{eff} ($\alpha = 0.5$) and $\lambda = 0.5$, on every backbone and benchmark. In the tabular model of this appendix, the per-sample contrastive logit gradient has norm $\beta/\sqrt{2}$, while the anchor gradient $\lambda\|\pi_{\mathrm{ref}} - \delta_{a^*}\|$ can be up to $2\lambda/\beta$ times larger. This comparison does not describe the implementation, for three reasons. (i) The anchor is averaged over the action tokens, whereas the pair log-probabilities are summed over them. (ii) Per token, the anchor's gradient scales with $1 - \pi_\theta(a_t^+ \mid \cdot)$ and is small after the warm-up except at the few tokens where the student disagrees with the expert, whereas the gradient of the (near-linear) push-down of $\log \pi_\theta(a^-)$ does not shrink as the negative becomes unlikely. (iii) Adam normalizes each coordinate by its running gradient scale. Measured on τ-bench with Qwen3.5-9B, the contrastive term accounts for 38% of the norm of the Adam update (median over steps; Appendix F), which is why a small β still has a large effect and why test accuracy is flat for $\beta \geq 2 \times 10^{-3}$ but falls off toward $\beta = 10^{-4}$ (Table 8).

M CONSTRUCTION OF THE STABLETOOLBENCH-PERTURBED EVALUATION SET

The perturbed StableToolBench columns in Tables 3, 4, 10, and 35 are evaluated on a perturbed variant of StableToolBench. This appendix documents its construction.

Perturbation classes. We define nine perturbation operators that act on a StableToolBench instance (system prompt + tool list + user query) without changing the latent decision needed to answer the query. Each operator is parameterized by a difficulty level $L \in \{1, 2, 3\}$; a higher level applies a more aggressive transformation. Concretely, the nine operators are:

1. `rename_tool` – rename each tool's `name` field. L1: case changes only; L2: synonym substitution; L3: aggressive rewrite.
2. `rename_fields` – rename argument fields. L1: case changes; L2: synonyms; L3: synonyms plus decoy fields the model must ignore.
3. `reorder_args` – shuffle argument order in the tool spec. L1: shuffle optional fields; L2: shuffle all fields and reverse tool list order; L3: merge required/optional and shuffle.
4. `format_switch` – reword the system prompt and tool spec. L1: minor wording; L2: style shifts and restructure; L3: convert tool spec to OpenAI function-calling JSON, YAML, or prose.
5. `verbosity` – add filler context. L1: one extra sentence; L2: roughly double-length redundant context; L3: roughly five-times longer with misleading capability claims.
6. `info_position` – move the relevant information within the prompt. L1: middle of the prompt; L2: end of the prompt with a long preamble; L3: bury the relevant information among unrelated content.
7. `distractor` – add unrelated tools. L1: 3 distractors; L2: 5 distractors from same/other categories; L3: 8 distractors with name similarity to the correct tool.
8. `desc_degrade` – shorten or remove descriptions. L1: drop optional-argument descriptions; L2: drop all argument descriptions; L3: drop tool and argument descriptions.
9. `alias_rename` – rename to opaque aliases. L1: light synonyms; L2: domain aliases; L3: opaque IDs (e.g., `tool_17`, forcing pure description-based selection).

The set is generated by the script in our code release with a single fixed seed `seed`$=42$, and stored as one JSON file per (operator, level, original subset) combination. The full Cartesian product yields $9 \times 3 \times 6 = 162$ files (the six original StableToolBench evaluation subsets G1_category / G1_instruction / G1_tool / G2_category / G2_instruction / G3_instruction). The aggregate Pert. number reported in the main text is the unweighted mean over the level-2 subset of these files, which we found gives a representative middle-of-distribution perturbation difficulty.

Relation to training-time PPA. On StableToolBench, Agentic-DPO is trained with the four action-rendering views described in Appendix N (`base`, `json`, `rename`, `combined`) and with recovery-context views. Two evaluation operators overlap in spirit with these training views (`rename_tool` matches `rename`; `format_switch` matches `json`/`combined`); the other seven (`rename_fields`, `reorder_args`, `verbosity`, `info_position`, `distractor`, `desc_degrade`, `alias_rename`) target classes outside training-time PPA. At level 2, `alias_rename` (domain aliases) is close in spirit to the `rename` view, so the separation between the two groups is approximate.

Per-operator results and levels. Table 38 breaks the perturbed column of Table 3 down by operator for Qwen3.5-2B (the column is the unweighted mean of the nine level-2 operators) and reports levels 1 and 3 and the mean without the two operators that resemble training views. Agentic-DPO leads PPA+SFT by 3.3 to 4.6 points on each of the seven other operators, but ETO and PivotRL are as robust or more robust on most of them: Agentic-DPO's lead over these baselines on the perturbed set comes from the two operators that resemble training views. Per-operator standard deviations over seeds are 1.6–3.4 points.

Table 38: Accuracy per perturbation operator on StableToolBench with Qwen3.5-2B (means over ten seeds).

Operator (level 2)	SFT	PPA+SFT	ETO	PivotRL	Agentic-DPO
`rename_tool`	49.8	85.9	86.9	86.6	88.6
`rename_fields`	51.2	80.2	84.6	84.9	84.1
`reorder_args`	55.9	83.5	86.8	86.3	87.2
`format_switch`	47.3	84.6	86.1	85.8	88.0
`verbosity`	54.1	80.8	84.3	84.6	84.5
`info_position`	53.6	79.9	83.9	83.5	83.2
`distractor`	50.2	77.3	82.7	82.1	81.9
`desc_degrade`	55.0	78.9	84.0	83.6	82.6
`alias_rename`	56.3	81.5	85.7	86.7	84.9
Level 2, all nine (main tables)	52.6	81.4	85.0	84.9	85.0
Level 2, without the two PPA-like operators	53.8	80.3	84.6	84.5	84.1
Level 1, all nine	55.0	82.9	86.5	86.4	88.1
Level 3, all nine	44.9	74.2	78.1	77.9	79.6

N Policy-Preserving Augmentation: Worked Examples

This appendix gives worked examples of the action-rendering views on τ-bench retail and Mind2Web and of the recovery-context view, which is used only in the tool domains (StableToolBench and τ-bench).

N.1 Action-Rendering Variants

For each canonical pair (s_t, a_t^+), we draw a view $\phi \sim \mathrm{Unif}(\Phi)$ from a four-element set $\Phi = \{\texttt{base}, \texttt{json}, \texttt{rename}, \texttt{combined}\}$. The rendered state s_t^ϕ uses the corresponding tool spec and chat format; the rendered expert action $a_t^{+,\phi}$ is the same latent decision under the chosen schema. We illustrate by rendering one canonical τ-bench retail pair and one canonical Mind2Web pair under all four views.

τ-bench retail example. Canonical pair: expert tool `find_user_id_by_name_zip` with arguments `{"first_name":"Yusuf","last_name":"Rossi","zip":"19122"}`.

- **base** (ReAct, canonical tool names):

```
Action:
{"name": "find_user_id_by_name_zip",
 "arguments": {"first_name": "Yusuf", "last_name": "Rossi",
               "zip": "19122"}}
```

- **json** (Hermes-style `<tool_call>` tags, canonical tool names):

```
<tool_call>
{"name": "find_user_id_by_name_zip",
 "arguments": {"first_name": "Yusuf", "last_name": "Rossi",
               "zip": "19122"}}
</tool_call>
```

- **rename** (ReAct, renamed tool from `TOOL_RENAME_MAP`):

```
Action:
{"name": "lookup_customer_by_name_zip",
 "arguments": {"first_name": "Yusuf", "last_name": "Rossi",
               "zip": "19122"}}
```

- **combined** (Hermes `<tool_call>` tags + renamed tool):

```
<tool_call>
{"name": "lookup_customer_by_name_zip",
 "arguments": {"first_name": "Yusuf", "last_name": "Rossi",
               "zip": "19122"}}
</tool_call>
```

The state messages and tool spec headers used by the system prompt are rewritten consistently with each view, so the `rename` and `combined` variants present a tool spec with the renamed tool names; the model never sees the canonical name during a `rename`-rendered step.

Mind2Web example. Canonical pair: expert operation `SELECT` with arguments `{"choice":"C","value":"Pickup"}`.

- **base** (ReAct, canonical operation names):

```
Action:
{"name": "SELECT",
 "arguments": {"choice": "C", "value": "Pickup"}}
```

- **json** (Hermes-style `<tool_call>` tags, canonical operation names):

```
<tool_call>
{"name": "SELECT",
 "arguments": {"choice": "C", "value": "Pickup"}}
</tool_call>
```

- **rename** (ReAct, renamed operation):

```
Action:
{"name": "pick_option",
 "arguments": {"choice": "C", "value": "Pickup"}}
```

- **combined** (Hermes `<tool_call>` tags + renamed operation):

```
<tool_call>
{"name": "pick_option",
 "arguments": {"choice": "C", "value": "Pickup"}}
</tool_call>
```

Rename maps. The renaming used by the `rename` and `combined` variants is a single fixed dictionary per benchmark, applied consistently across the tool spec and the expert action so that the latent decision is preserved.

- τ-bench retail: `TOOL_RENAME_MAP` renames the 16 retail tools (the dialog-only `respond` action is never renamed). A 5-of-16 slice:

```
find_user_id_by_email        -> lookup_customer_by_email
find_user_id_by_name_zip     -> lookup_customer_by_name_zip
get_user_details             -> fetch_user_profile
get_order_details            -> fetch_order_info
get_product_details          -> fetch_product_info
```

- Mind2Web: the operation set is small (3 entries), and the rename is total:

```
CLICK  -> tap
TYPE   -> input_text
SELECT -> pick_option
```

Applying PPA to a new domain. PPA requires no semantic relabeling of the data. It applies consistency-preserving transformations to parsed actions, and porting it to a new agent environment takes three components that most agent codebases already contain. First, a parser that maps an action string of the existing schema to a latent action (tool name plus normalized arguments, or operation plus target), which is also the parser used to filter expert-equivalent candidates in Section 3.2. Second, one renderer per schema view, each of which writes a latent action and the tool specification in the state under that view; the views used here are a ReAct-style call and a tagged JSON call. Third, a fixed rename map from canonical tool names to alternative names, applied to the specification and to the action string together. The recovery-context view needs the list of the domain's read-only tools, from which an earlier call and an error or empty observation are injected. The transformations are released with the code.

N.2 RECOVERY-CONTEXT VARIANT

The recovery-context variant inserts an erroneous earlier action and its correction into the trajectory's `state_messages` before a randomly chosen expert step. The corrected step then becomes the positive at training time, simulating the situation in which the agent had previously taken a wrong action and must recover. The injected action is always a read-only tool from the trajectory's own toolset (never a hallucinated name), paired with an error or empty observation.

Worked example (state messages slice). Suppose the canonical trajectory contains the slice:

```
[2] assistant: Action: find_user
                Action Input: {"name": "Alice"}
[3] user:      Observation: {"user_id": "123"}
[4] assistant: Action: get_order
                Action Input: {"user_id": "123"}
```

After applying recovery-context PPA at index 4, the trajectory becomes:

```
[2] assistant: Action: find_user
                Action Input: {"name": "Alice"}
[3] user:      Observation: {"user_id": "123"}
[4] assistant: Thought: Let me try using get_product...   [INJECTED]
                Action: get_product
                Action Input: {"user_id": "123"}
[5] user:      Observation: {"error": "Tool returned no results"}
                                                          [INJECTED]
[6] assistant: Thought: The previous tool call failed. Let me try
                a different approach.                     [RECOVERY]
                Action: get_order
                Action Input: {"user_id": "123"}
```

Indices 4 and 5 are the injected mistake and its observation; index 6 is the original expert action with a prepended recovery thought, now serving as the expert positive at this step. The latent decision (call `get_order` with the user id from the previous step) is preserved.

Conditions for policy preservation. The recovery-context view is policy-preserving only if the injected step changes neither the environment state nor the information available at the current step. We therefore inject only tools that the domain marks as read-only (lookups and searches) and pair them with an error or empty observation, and we never inject a step before an expert action that depends on the injected tool's output. Replaying 500 augmented τ-bench contexts in the simulator left the database state unchanged in all cases, and the expert's next action remained executable and correct in all of them.

Why this is policy-preserving. The decision the model is asked to make at the rendered step (call `get_order` on `user_id: 123`) is identical in the canonical and recovery-context views. The difference is only in the conversation history that precedes this state. Because the injected call is read-only and returns an error or empty result, the recovery-context view changes neither the environment state nor the information available at the current step, so the set of valid expert actions is unchanged; it only changes the context the model conditions on.