NEGATIVES SHOULD MATCH THE STUDENT: TEACHER–STUDENT–JUDGE PREFERENCE DISTILLATION FOR AGENTIC TOOL USE

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-weight language models lag behind frontier systems on "agentic" workflows: long-horizon planning, strict tool-schema adherence, and policy compliance under adversarial user pressure. We present the **Teacher–Student–Judge (TSJ)** pipeline, which synthesizes Direct Preference Optimization (DPO) pairs by letting the target model (Student) compete against a stronger Teacher inside simulated multi-turn conversations, with a Judge that selects winners and synthesizes negatives against the Student's own successful replies. Fine-tuning Qwen3-30B-A3B on 7,747 TSJ pairs improves τ^2-bench Telecom from 0.04 to 0.13 and Retail from 0.58 to 0.64, a $+6.4$-point macro-average gain whose task-level bootstrap interval excludes zero (Airline $+0.03$, within noise), and cuts Telecom episodes aborted for calling nonexistent tools from 67–75 of 114 to 1–2; general capabilities remain within 1.3 points across four evaluations (MMLU, GSM8K, and HumanEval improve, while IFEval declines 0.9 points). **Attribution:** in four-condition ablations on Qwen3-4B, a size-matched control that keeps Judge-synthesized negatives scores 3.4 points above distillation-only (directional), and the full dataset 4.7 points above it (interval excludes zero). **Generality and its boundary:** when Granite-4.1-8B is trained on Qwen-generated data the gain is inside noise ($+2.0$ points); regenerating the data with Granite as Student gives $+3.8$, and at identical dataset size Granite-matched synthesized negatives beat negative-free distillation by +4.5 points (interval excludes zero). **Offline vs. online:** at matched scale, offline TSJ-DPO matches online GRPO trained directly in the executable τ^2-bench environment (0.432 vs. 0.419 macro-average). These results position student-anchored preference pairs, rather than generic teacher demonstrations, as the highest-leverage data for agentic preference optimization.

1 INTRODUCTION

The rapid evolution of LLMs has shifted focus from chat toward "agentic" behavior: autonomously planning, executing tool calls, and adhering to business logic over multi-turn conversations (Wang et al., 2024). Frontier proprietary models handle this well; open-weight models in the 4–30B range systematically fail through context drift, premature tool execution, and hallucinated parameters. We call these failures the *agentic gap*.

Supervised fine-tuning (SFT) rarely closes this gap because its maximum-likelihood objective has no contrastive term that penalizes *policy* violations (SFT corpora can of course contain error-recovery demonstrations (Chen et al., 2024); what they cannot express is a per-turn preference between two candidate replies) ("you called the tool correctly, but forgot to verify identity first"). Preference optimization can carry that signal, but DPO (Rafailov et al., 2023) is only as good as its pairs (x, y_w, y_l). The obvious recipe (sample a frontier teacher for y_w and the base model for y_l) raises an equally obvious objection: *is anything gained beyond distilling a stronger teacher?* A second question follows from the benchmarks themselves, which ship executable environments: *why not simply run online RL with verified rewards?*

This paper answers both questions empirically. Our contributions:

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

1. **The Teacher–Student–Judge (TSJ) pipeline** (Section 3): a synthetic-data architecture in which a Judge does more than select between Student and Teacher: when the Student wins a turn, the Judge *synthesizes* a plausible policy-violating alternative as the rejected sample, anchoring preference pairs to the Student's own behavior distribution. Applied to Qwen3-30B-A3B, 7,747 TSJ pairs deliver +6/+9 points on τ^2-bench Retail/Telecom with no large general-capability regression across four evaluations (Section 5).
2. **Attribution via controlled ablation** (Section 6): four conditions (base, full TSJ, distillation-only with Judge negatives removed, and a size-matched control) separate the Judge mechanism from teacher quality and data quantity, across two model families.
3. **The student-matching principle** (Section 7): across all four (student, dataset) pairings, the recipe delivers its gain only when the negatives were generated against the model being trained — including a control in which the *Qwen* student is trained on Granite-native data and the full recipe gains nothing ($+0.6$ points, inside noise), which argues against "the second dataset is simply better". A training-dynamics analysis (Section 8) offers the best-supported account of the failure mode: teacher-vs-student pairs alone admit a shortcut solution, a style discriminator, which student-anchored synthesized negatives break.
4. **Offline–online comparison** (Section 9): under a shared held-out protocol at 4B scale, offline TSJ-DPO matches online GRPO trained in the executable τ^2 environment, addressing the strongest alternative baseline directly.

Alongside the pipeline we will release both TSJ datasets as standalone artifacts: 7,747 preference pairs generated against Qwen3-30B-A3B and 10,137 pairs generated against Granite-4.1-8B, each with per-pair provenance metadata (winner source and judge issue labels). Because the two datasets were produced by disjoint model ensembles, they double as the experimental instrument behind Sections 7 and 8.

2 Related work

Synthetic data for alignment. *Magpie* (Xu et al., 2024) demonstrated that aligned LLMs can synthesize instruction data without seed prompts, but open-ended generation under-covers business-logic edge cases. Our pipeline instead uses a targeted *User Simulator* to generate adversarial scenarios, similar in spirit to agent-tuning corpora (Zeng et al., 2024). *Agent-FLAN* (Chen et al., 2024) also embeds negative behaviour in *supervised* data by interleaving error and recovery trajectories; the difference is that TSJ supplies a contrastive preference signal whose negatives are generated against the target model, rather than demonstrations of failure shared by all students. Unlike *DiaTool-DPO* (Jung et al., 2025), which focuses on general dialogue states, TSJ targets policy-boundary failures, and unlike selection-only judging, our Judge synthesizes negatives conditioned on the Student's winning reply. Preference-based self-distillation (Yu et al., 2026) likewise keeps student sampling on-policy during distillation; TSJ instead anchors *offline* pairs to the student's own replies.

Preference optimization variants. DPO (Rafailov et al., 2023) aligns models by contrasting preferred and dispreferred responses without an explicit reward model; IPO (Azar et al., 2024), KTO (Ethayarajh et al., 2024), and ORPO (Hong et al., 2024) vary the objective or relax pairing requirements. We use standard DPO throughout: our contribution concerns *pair construction*, and our ablations show pair provenance dominates outcome quality at fixed objective. Closest to our claim, Tajwar et al. (2024) show that preference data drawn from the policy's own distribution (on-policy, even when suboptimal) beats off-policy data of higher quality; the student-matching principle is that finding carried to judge-synthesized negatives anchored to the student. Pan et al. (2025) argue that chosen-response quality dominates DPO while rejected-response quality matters little; our Judge-negative ablations (Section 6) test that claim in the agentic setting.

Online RL with verifiable rewards. When an executable environment provides deterministic rewards, GRPO (Shao et al., 2024) and PPO-family methods are the natural baseline; τ-bench (Yao et al., 2024) and τ^2-bench (Barres et al., 2025) offer such environments. Rather than argue offline methods are exempt from the comparison, we run it (Section 9).

Tool-augmented language models. Toolformer (Schick et al., 2023), ToolLLM (Qin et al., 2023), Gorilla (Patil et al., 2024) and API-Bank (Li et al., 2023) scale tool-use training and evaluation, largely for single-turn selection accuracy. Specialist models such as xLAM optimize function-calling

syntax, which the alignment-tax hypothesis predicts may trade off against multi-turn policy adherence (Ouyang et al., 2022); on our harness the specialist xLAM-2-32b in fact scores at or above our 30B model (Section 5), so we make no such claim.

3 THE TEACHER–STUDENT–JUDGE PIPELINE

TSJ is a mechanism for synthesizing student-anchored preference pairs under explicit behavioral constraints. Five model roles cooperate (diagram: Appendix C):

- **Scenario Generator (M_{scen}):** samples from a weighted library of 34 "hard" interaction templates (Appendix L) and emits a structured scenario: assistant directives, user persona, tool schemas, safety notes, and a scenario-specific system prompt used as the sole runtime system message.
- **User Simulator (M_{user}):** plays the caller across turns, injecting pressure, false premises, mid-conversation goal changes, and noisy phrasing; it also simulates tool execution results.
- **Student ($M_{student}$):** *the model that will be trained.* At every assistant turn it produces a candidate reply against the identical context as the Teacher.
- **Teacher ($M_{teacher}$):** a stronger reasoning model producing the reference reply.
- **Judge (M_{judge}):** compares both replies under a strict checklist (temperature 0.1; prompt in Appendix K). If the **Teacher wins** and the Student's failure is substantive (factual, tool-misuse, or safety-relevant), the pair (Teacher $= y_w$, Student $= y_l$) is kept; this records a *real* student failure. If the **Student wins**, the Judge *synthesizes* a short free-text alternative containing a plausible failure as y_l, keeping the Student's own reply as y_w. Two cases yield no training pair: a Teacher win whose issue tags are only stylistic or minor, and a Student win for which the Judge returns no synthesized rejection. In both the turn is skipped, and in every case the winning reply is what gets appended to the conversation history.

The winning reply is appended to the conversation history and the interaction continues (up to 24 assistant turns), so a single conversation yields many pairs with shared prefixes: snapshots of the same trajectory at increasing depth.

We optimize with the standard DPO objective:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log \sigma\Big(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\Big)\Big] \tag{1}$$

The contribution is not the loss but the pair construction: both branches of the Judge anchor y_l to failures that are *reachable* from the Student's behavior distribution: either failures the Student actually produced, or Judge-written failures at turns the Student actually got right. We call these negatives student-anchored rather than "hard": under the untrained student the synthesized negatives are in fact easy to separate by likelihood margin, and what they contribute is a student-written y_w (Appendix I). One caveat is worth stating up front: replacing teacher-written negatives with judge-written ones substitutes one non-Student author for another, so it could in principle introduce a fresh stylistic cue rather than remove one. Section 8 tests this directly through held-out preference loss and response length.

3.1 TWO INSTANTIATIONS, TWO DATASETS

A central experimental resource is that we ran the pipeline *twice*, with disjoint model ensembles and different Students (Table 8, Appendix G; statistics in Appendix B): Dataset A targets Qwen3-30B-A3B (7,747 pairs, 29% synthesized), Dataset B targets Granite-4.1-8B (10,137 pairs, 14%). Every role is filled by a different model across the two, which guards against ensemble-specific effects; the scenario library, conversation loop, judge checklist and DPO schema are shared.

The teacher-win share differs because Granite-4.1-8B loses to its teacher more often than the 30B student did; the synthesized-negative fraction is a property of the student's strength, not a free parameter.

3.2 SCENARIO DESIGN

Scenarios are sampled from 34 weighted variants, grouped for reporting into tool-parameter precision (about half of Dataset A's sampling weight), complex task resolution (about a third), targeted airline/retail flows ($\sim$12%) and misinformation/safety ($\sim$3%; Appendix L). Dataset B raises the airline/retail share to $\sim$30% and diversifies the system-prompt register (52% voice-framed, 48% channel-neutral).

4 EXPERIMENTAL SETUP

Models and training. All training uses LoRA DPO with $\beta = 0.15$, lr 2×10^{-6} (cosine), sigmoid loss, and bfloat16. The retained configuration for the reported 30B DPO continuation uses LoRA r=16/α=32 on 8×H100 (checkpoint-1410 selected on dev loss); the 4B and 8B ablation models also use r=16/α=32, 3 epochs, effective batch 16, on single-GPU workstations. Within each ablation, *every* hyperparameter is held fixed across conditions; only the training slice changes. Granite runs on Dataset B use max sequence length 10,240 (fitted to that dataset's longer conversations); this too is constant across its conditions.

Judge validation. We manually audited 100 randomly sampled Judge decisions from Dataset A (50 Student wins, 50 Teacher wins) against one expert annotator's own choice of the better reply: agreement was 94% (47 and 47 confirmed, 3 disagreements each way; Cohen's $\kappa = 0.88$; single annotator), suggesting the checklist protocol mitigates self-preference bias. We additionally categorized 100 sampled *synthesized* negatives (Appendix K): 47% are the policy-violating alternatives the pipeline is designed to produce, 17% are unjustified refusals, 31% differ from the accepted reply mainly in tone or length, and 5% carry a formatting artifact. Counting refusals as behavioral failures (the Judge prompt names them as a target), 64% encode a behavioral failure and 36% differ only in tone or format; counting only the intended policy violations, just over half (53%) are off-target. Either way this is a real limitation of the current Judge prompt, reported as measured.

Evaluation protocols. Three distinct protocols are used, and *absolute numbers are comparable only within a protocol*:

- **P1 (flagship, 30B):** τ^2-bench v1.0.1 airline (50 tasks), retail (114), telecom (114), ACE-Bench, and BFCL, n=3 evaluation runs; the evaluated model is the agent, served by vLLM 0.25.1 with a 16,384-token context, and GPT-4.1 is the user simulator.
- **P2 (ablations, 4B and 8B):** τ^2-bench (Barres et al., 2025) airline (50 tasks), retail (114), telecom (114), 3 seeds, GPT-4.1 user simulator, vLLM with the model-correct tool-call parser. A base-model *sanity gate* halts runs showing parser-failure signatures (Appendix A.4).
- **P3 (offline vs. online, 4B):** per-domain train/held-out splits (airline 30/20, retail 68/46, telecom 68/46, seed 42), 3 seeds on held-out tasks only, DeepSeek-V4-Flash user simulator for training rollouts and for evaluating *all* arms (its lower cost makes the online rollouts affordable).

Conditions (P2). *base* = untrained; *full* = all pairs; *distill* = teacher-win pairs only (Judge-synthesized negatives removed, exactly the "naive distillation" alternative); *control* = a seeded random subset of *full*, size-matched to *distill*. Because *control* and *distill* have identical size and source, their difference isolates the presence of synthesized negatives; *full* − *control* isolates data quantity at fixed composition.

5 FLAGSHIP RESULTS: QWEN3-30B-A3B

Table 1 reports protocol P1. TSJ-DPO lifts Retail from 0.576 to 0.637 and Telecom from 0.035 to 0.132, a macro-average gain of $+0.064$ (95% interval $+0.022$ to $+0.105$, positive in every seed); the telecom gain is significant on its own ($+0.050$ to $+0.149$), the retail gain is directional (-0.015 to $+0.137$). ACE-Bench and BFCL also improve. Airline improves by $+0.033$ with an interval of -0.047 to $+0.113$, so we do not claim an airline effect: that domain is the noisiest in our evaluation and we return to its variance in Section 10. These are gains over the model's own base, not a system

Table 1: Flagship results (protocol P1, 3 seeds; ± = half-width of a task-level bootstrap 95% interval, Appendix A), measured under τ^2-bench v1.0.1 at a 16,384-token serving context. Published figures for external models are omitted: re-measuring xLAM-2-32b on our harness moved its retail score from 0.41 to 0.63, so cross-paper τ^2 numbers are not comparable.

Model	**Airline** (τ^2)	**Retail** (τ^2)	**Telecom** (τ^2)	**ACE Bench** (Reasoning)	**BFCL** (Syntax)
Qwen3-30B-A3B-Instruct	0.333±0.107	0.576±0.076	0.035±0.025	0.615	34.11%
Ours (TSJ-DPO)	0.367±0.097	**0.637±0.069**	**0.132±0.048**	0.661	37.16%

Table 2: Four-condition ablation on Qwen3-4B (Dataset A, protocol P2, 3 seeds; ± = half-width of a task-level bootstrap 95% interval; macro-average over domains). *distill* = Judge negatives removed; *control* = random subset of *full*, size-matched to *distill* (5,490 rows each). Per-domain CIs overlap between adjacent conditions; the macro-average ordering, not any single domain, carries the attribution claim.

Condition	**Airline**	**Retail**	**Telecom**	**Avg**
base	0.307 ± 0.110	0.443 ± 0.070	0.123 ± 0.050	0.291
distill (no synth. neg.)	0.327 ± 0.107	0.447 ± 0.079	0.140 ± 0.054	0.305
control (size-matched)	0.360 ± 0.113	0.482 ± 0.076	0.173 ± 0.057	0.338
full (TSJ)	**0.380 ± 0.120**	**0.518 ± 0.079**	0.158 ± 0.054	**0.352**

ranking: re-measured on the same harness, the function-calling specialist xLAM-2-32b reaches macro 0.405 against our 0.379, a difference inside its interval (Appendix A).

Capability preservation. Standard benchmarks move within 1.3 points: MMLU 77.38 → 77.92, GSM8K 80.44 → 81.73, HumanEval 93.29 → 93.90, IFEval 82.62 → 81.70 (5-shot/standard). No general-capability regression drives the agentic gains.

5.1 Multi-trial robustness: pass^k

Single-trial pass rates conflate reliable policy execution with lucky rollouts. We therefore re-evaluated the flagship and its base under τ^2-bench with $K{=}4$ independent trials per task and report pass^k, the probability that all of k randomly drawn trials succeed (Appendix D). TSJ-DPO is at least as good as base at every k in every domain. On airline the margin *widens* with k, from $+0.025$ at $k{=}1$ to $+0.060$ at $k{=}4$: the gain is in repeatability, not first-trial luck. On retail, pooling two 4-trial samples per model gives pass^8 0.263 vs. 0.214, a $+23\%$ relative margin.

Telecom is a special case worth reporting. The base model frequently calls *user-side* device tools (e.g. `check_sim_status`) as if they were agent tools; τ^2 returns "tool not found" and aborts the episode after ten such errors, which happens in 67–75 of 114 base episodes per seed versus 1–2 for TSJ-DPO, which far more often instructs the user to perform the check instead. The verify-then-act behavior the pipeline trains for is thus visible as an aborted-episode statistic, not only as a pass rate.

6 Attribution: what drives the gains?

The flagship result admits a deflationary reading: *any* preference pairing against a 1T-parameter teacher might produce it. The four-condition design (Section 4) tests this directly at a scale where controlled retraining is affordable, on Qwen3-4B-Instruct-2507 with Dataset A.

Table 2 isolates three factors: teacher distillation, exposure to Judge-synthesized negatives, and dataset size. Comparing *full* with *distill* tests the combined contribution of the synthesized-negative slice and its additional rows; *control* with *distill* holds row count fixed; and *full* with *control* isolates the contribution of additional data at fixed composition. **Naive distillation is not enough:** removing the Judge-synthesized slice costs 4.7 macro points (0.352 → 0.305, interval $+0.6$ to $+8.7$), though it also cuts the dataset from 7,747 to 5,490 rows, and *distill* retains every teacher-written chosen reply.

Table 3: The student-matching grid: macro-average over airline/retail/telecom for every (student, dataset) pair we ran, with the two mechanism contrasts. **All four cells were measured under one configuration** — τ^2-bench v1.0.1, vLLM 0.25.1, full 50/114/114 task sets, a 16,384-token serving context and three seeds — so absolute values are comparable across rows as well as within them. Each family shares one base measurement. Deltas are computed from unrounded seed-level means; bold deltas have a task-level bootstrap 95% interval excluding zero. Per-domain values: Appendix E.

Student	Training data	Match	base	distill	control	full	full−base	ctrl−dist
Qwen3-4B	A (Qwen-native)	yes	0.291	0.305	0.338	**0.352**	**+0.061**	+0.034
Granite-4.1-8B	B (Granite-native)	yes	0.292	0.304	0.348	**0.331**	+0.038	**+0.045**
Granite-4.1-8B	A (Qwen-native)	no	0.292	0.276	0.262	0.312	+0.020	−0.014
Qwen3-4B	B (Granite-native)	no	0.291	0.320	0.323	0.297	+0.006	+0.003

At matched size, *control* scores 3.4 points above *distill* (0.338 vs. 0.305, interval -0.4 to $+6.9$, so directional rather than significant); since *control* is sampled from *full*, this reflects synthesized-negative exposure and the resulting composition together. **Quantity adds little** (*full* − *control* $= +1.4$, inside its interval) and **teacher distillation alone carries little** (*distill* − *base* $= +1.4$, inside its interval): on Qwen the pairs' *negative* side carries signal beyond the teacher's positives.

7 THE STUDENT-MATCHING PRINCIPLE

Does the recipe transfer across model families? We repeated the identical four-condition ablation on **Granite-4.1-8B**, a dense, agentic-tuned model from a different lineage, twice: once trained on Dataset A (whose negatives encode the *Qwen* student's failures) and once on Dataset B (regenerated with *Granite itself* as Student, under a disjoint model ensemble; Appendix B). Table 3 collects every (student, dataset) cell.

With mismatched data, the recipe does not deliver. On Dataset A, *full* beats base by only +2.0 points (interval -1.9 to $+5.7$), and the size-matched *control* — the condition that isolates synthesized-negative exposure — is the *worst* of the four (0.262 vs. base 0.292). Mismatched synthesized negatives are not merely inert: at fixed row count they cost performance (directional; the control−distill interval spans zero).

With student-matched data, everything flips. On Dataset B, *full* gains +3.8 points over base (interval -0.3 to $+8.0$, positive in every seed), and both mechanism deltas turn positive: at *identical size and source*, pairs containing Granite-matched synthesized negatives beat negative-free distillation by **+4.5 points** (interval $+0.9$ to $+8.2$, positive in every seed). Appendix E reports all per-domain outcomes and identifies the corresponding contrast for each headline delta.

Testing "Dataset B is simply better data". Datasets A and B differ in the Student *and* the generating ensemble, so Granite's success on B could reflect a better ensemble. Training the *Qwen* student on Dataset B (mismatched for it) under the identical protocol separates the two: Qwen gains $+0.006$ macro over base ($0.291 \rightarrow 0.297$, interval -0.030 to $+0.042$). Against the $+0.061$ it gains from its *own* Dataset A (interval $+0.028$ to $+0.097$), the benefit of the *full recipe* tracks the match, not the dataset. Dataset B's teacher-written positives are if anything better for Qwen (*distill* $+2.9$ points, against $+1.4$ from Dataset A); what fails to transfer is the negative side, which takes Qwen from 0.320 back to 0.297.

The behavioral signature. Macro-averages differ by a few points between these cells, but the behavior the pipeline targets separates them cleanly: count telecom episodes τ^2 terminates after ten "tool not found" errors (Section 5). With matched Dataset B, *control* and *full* abort 1–6 episodes of 114 against base 60–70 and *distill* 26–36; with mismatched Dataset A the same conditions abort 46–78, i.e. no better than base. Per-seed counts: Appendix J.

With matched data the negatives all but eliminate the failure, and half of them suffice (*control* matches *full*); with mismatched data the same manipulation leaves *control* in the base band. The distributions do not overlap — a sharper separation than any pass-rate contrast here — and the same ordering holds for the 30B flagship (67–75 versus 1–2) at both serving contexts we tested.

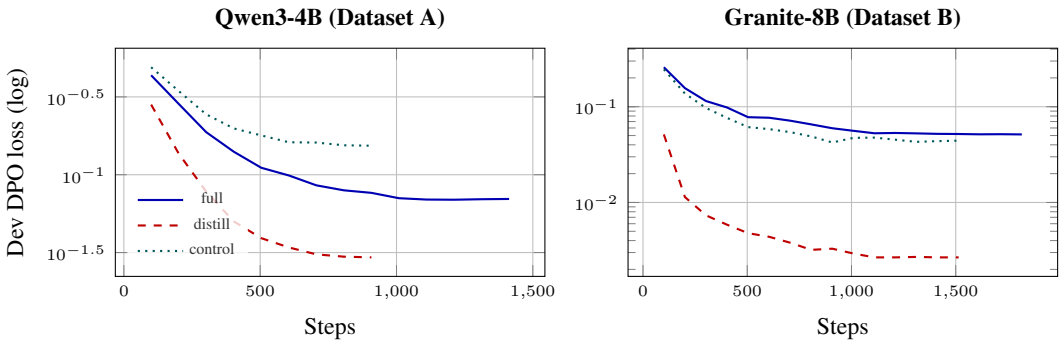

Figure 1: **The style-shortcut fingerprint: the lowest DPO loss transfers worst.** Held-out DPO loss (log scale). In both families, *distill* (dashed) collapses far below the synth-negative conditions (Granite: 0.003 vs. 0.044/0.051; Qwen-4B: 0.029 vs. 0.070/0.154), yet transfers *worst* to the benchmark. Near-perfect pair separation with poor downstream transfer indicates the pairs are separable by a shallow feature.

Because Dataset B was produced by a completely different Teacher/Judge/Simulator ensemble, and because the Qwen-on-B cell removes the ensemble-quality explanation, the flip cannot be attributed to any particular teacher, judge, or simulator. What changed is *whose failures the negatives encode*. The design does not rule out an interaction between a particular Judge's writing and a particular student; a third dataset with Dataset A's ensemble and Granite as Student would isolate it, and we did not run one. These experiments support what we propose as a student-matching principle, in line with the on-policy advantage reported by Tajwar et al. (2024): **TSJ negatives are most effective when they are derived from the deployment model's own failure surface.** A practical corollary: a TSJ dataset is not a portable artifact *across model families*; the pipeline is the artifact. Within a family the constraint is looser: Dataset A was generated against Qwen3-30B-A3B and still trains Qwen3-4B, so re-generation is required per family, and per model only when the failure surface differs materially.

Domain-level caveat. On Dataset B, *control* beats *full* on retail, consistent with the extra teacher-win rows carrying the style shortcut (Section 8).

8 Why mismatched negatives fail: the style-shortcut effect

The training dynamics expose a mechanism for both the *distill* deficit and the mismatch failure. Figure 1 plots held-out DPO loss for the three trained conditions in both ablation studies. The held-out 3% was sampled per row, not per conversation, so a held-out row's prefix overlaps training rows from the same conversation while its (chosen, rejected) pair does not; absolute loss values are therefore optimistic, and the argument below uses only the contrast between conditions, which the overlap affects alike. Counterintuitively, the condition with the lowest held-out DPO loss is the worst-performing trained condition on the downstream benchmark, so low preference loss is not evidence of transferable agentic behavior.

Distillation pairs admit a shortcut. Every *distill* pair opposes *teacher-written* text (y_w) to *student-written* text (y_l). Two different models are stylistically distinguishable, so the DPO objective can be driven to near zero by learning a *style discriminator*: a shortcut that distinguishes teacher-written from student-written text through superficial distributional cues rather than task-relevant correctness, tool use, or policy compliance. The fingerprint is exactly what we observe: *distill* reaches dev loss 0.003 (Granite) / 0.029 (4B), an order of magnitude below the mixed conditions, while producing the worst benchmark transfer of the trained conditions (and eerily deterministic behavior, e.g., identical airline scores across all three seeds).

Synthesized negatives break the shortcut. In the Judge-synthesized pairs, y_w is the *student's own* reply and y_l is judge-written. On these rows the style heuristic gives the *wrong* answer, forcing the model to discriminate on content: which tool, which arguments, whether verification preceded action. Mixing only $\sim$14% such rows (Dataset B *control*) keeps dev loss an order of magnitude higher and

yields the +4.5 transfer gain. These rows also anchor y_w in the policy's own high-likelihood region, reducing the distribution shift that pure teacher-styled positives induce.

This also explains the mismatch failure. For Granite trained on Dataset A, *both* sides of every pair are out-of-distribution text (Qwen student, Kimi teacher); the synthesized negatives are failures written against *Qwen's* winning turns, targeting failure modes Granite does not have while missing the ones it does. The preference signal is spent policing another model's failure surface. At matched size, mismatched synthesized negatives trail negative-free distillation on Granite/Dataset-A (control $-$ distill $= -0.014$, control the weakest condition at 0.262) and are flat on Qwen/Dataset-B ($+0.003$); the two cells otherwise differ (on Granite/A *full* is the best trained arm, on Qwen/B *full* gains only 0.6), so we claim no single mechanism for mismatch beyond the shared result that the matched gain does not appear.

Neither length nor tone is the shortcut. Two shallow features could separate y_w from y_l instead of style: response length and tone. Appendix H shows that in the *distill* pairs the chosen reply is longer only 49% (Dataset A) and 32% (Dataset B) of the time, that the length cue in the synthesized pairs points the opposite way, and that adding the synthesized rows, 31% of which are tone-only, raises rather than lowers held-out loss, which is not what a tone discriminator would produce; we read this as evidence against a tone shortcut, not a demonstration, since a tone-filtered or length-matched ablation was not run.

Scope of the argument. A one-dimensional style heuristic is strictly self-contradictory only when Teacher and Judge coincide, as in Dataset A; Dataset B's three distinct authors leave a multi-class style ranking learnable in principle, and family-level formatting differences are part of what "failure surface" means here, so we present the mechanism as the best-supported account, not a proof (Appendix H).

9 Offline TSJ-DPO vs. online GRPO

Agentic benchmarks ship executable environments, making online RL with verified rewards the strongest alternative to any offline method. We run the comparison at 4B scale under protocol P3: per-domain task splits, training on the train split only, evaluation on held-out tasks, and one shared user simulator across all arms.

GRPO-Env trains Qwen3-4B directly in the τ^2 environment (8 iterations of vLLM rollouts, 4 episodes per task, group-relative advantages from the environment reward, clipped policy gradient on assistant tokens). **TSJ-DPO** is the *full* condition of Section 6, trained *offline* on Dataset A without access to τ^2's task instances or rewards. **GRPO-TSJ** runs GRPO in a synthetic environment built from the TSJ dataset itself (Appendix A).

Offline preference distillation matches online GRPO at 4B. TSJ-DPO attains the best macro-average (0.432; Table 7, Appendix F), edging GRPO-Env (0.419), which trains *inside the evaluation environment*, and above base (0.405); no macro contrast in this table is significant, and we claim parity, not superiority: an offline method without access to the environment's task instances or reward signal matches online RL with verified rewards, while remaining applicable when no executable environment exists. This differs from Lanchantin et al. (2025), who find that online and semi-online DPO and GRPO strongly outperform offline training on verifiable math and instruction following; their tasks are single-turn rather than multi-turn tool use, and given our overlapping CIs we report the disagreement rather than claim to resolve it. GRPO-TSJ under-performs base (0.359): reward signal harvested from static preference data did not survive the conversion to an online environment, an honest negative worth reporting: online methods need real environments, and offline methods need matched negatives. Its rule-based rewards are derived from the stored preference pairs, so an action that is valid but absent from the recorded trajectory scores as a miss; part of the gap is therefore attributable to reward rigidity rather than to the data, and we treat this arm as a lower bound.

What the comparison does and does not control. The arms are matched on parameters, protocol and user simulator, but not on *access to a stronger model*: TSJ-DPO inherits a frontier teacher's traces through Dataset A, GRPO-Env learns from its own exploration and the environment reward. Parity therefore says that offline data *carrying teacher signal* matches online RL at this scale, not that the offline algorithm equals the online one under equal supervision; a teacher-free offline arm would be needed, and we do not run it.

Airline regresses relative to base. On the held-out airline split, base scores 0.583 while TSJ-DPO reaches 0.483 and GRPO-Env 0.533: both trained arms lose ground on the domain the paper otherwise highlights. The TSJ-DPO regression is significant (interval -0.20 to -0.02) although the base number rests on 20 held-out tasks and carries an interval of ± 0.17; we flag the regression rather than report only the macro-average.

10 Discussion and limitations

What the student-matching principle implies. Evaluations of "synthetic preference data quality" that ignore student identity measure the wrong thing; within a family a shared failure surface, not an identical checkpoint, is what matters (Section 7). Regeneration is cheap: Dataset B cost roughly $220 of API inference.

Two composition limitations (Appendix I): synthesized negatives contain no tool calls, so every syntax-level negative sits in the shortcut-prone teacher-win slice (a plausible reason BFCL moves only $34.1\% \rightarrow 37.2\%$); and most training prefixes are teacher-written, an exposure-bias argument for raising the synthesized fraction, which is student-dependent (29% for the 30B, 14% for Granite-8B) and the clearest next sweep.

Where the gains sit, and where the noise is. Telecom is where the flagship gains most (0.035→0.132), and the gain is the aborted-episode effect of Section 5: the trained model stops calling tools it does not have. That is the behavior TSJ targets and our cleanest result, but it does not show that TSJ improves diagnostic hypothesis-testing, which our scenario library under-weights. **Airline is the exception in every protocol.** Under P1 the flagship gains $+0.033$ with an interval crossing zero; under P2 airline carries the widest interval of the three domains in every table; under P3 both trained arms regress on airline, TSJ-DPO by -0.10 (significant). Thirteen seeds per model give the same picture (9 of 13 wins, $+0.029$, interval crossing zero; Appendix A). The demonstrated wins are telecom and retail; on airline the method is at best neutral, and worse than base held-out. It is also the noisiest domain, and we rest no claim on it.

Breadth and baselines. Every multi-turn result is on τ^2-bench (ACE-Bench and BFCL are single-turn), and we do not compare empirically against Agent-FLAN or AgentTuning: both release SFT corpora and Llama-2 checkpoints, so a fair comparison requires retraining our students on their data, which we have not done.

Statistical scope. Every interval is a task-level paired bootstrap over 278 tasks and 3 seeds (Appendix A); "significant" means it excludes zero. Under that test the flagship gain, the Qwen full−distill and full−base contrasts, and the Granite control−distill contrast are significant; the other contrasts we discuss are directional.

Serving context. Every τ^2 number here uses a 16,384-token serving context: the 30B's training length, but 1.6–2× that of the 4B/8B ablations (8,192 / 10,240). Evaluating at 8,192 is not possible on this benchmark (airline's system prompt alone is $\sim$4,900 tokens; 70% of episodes exceed 8,192), so 16,384 is the smallest context it fits in. On the 30% of episodes that stay within 8,192 the condition ordering is unchanged and *full−base* is larger, not smaller. Scores depend on the serving context, so it should match the training length where the benchmark allows and be stated with every result.

Flagship training dynamics and qualitative case studies are in Appendices M and N.

11 Conclusion

We introduced the Teacher–Student–Judge pipeline and asked, across three models and two independently generated datasets, what does the work in a distillation-flavored result. The full recipe and the size-matched control both beat distillation-only, and a disjoint generator ensemble reproduces a positive result only when the data are regenerated against the target Student: what matters is whose failures the rejected responses represent. Code, data and weights will be released.

ETHICS STATEMENT

Synthetic data bias. Model-generated data can amplify generator biases. Our transactional scenarios (airline and retail) carry lower social-bias risk than open-ended dialogue, and the Judge prompt prioritizes substantive violations. However, style tags are common in Dataset A's synthesized negatives; the label breakdown in Appendix B makes this asymmetry explicit rather than claiming all negatives are policy-critical.

Agentic dual-use. Improving autonomous tool execution carries dual-use risk. Our training explicitly rewards verify-before-act behavior and resistance to social-engineering pressure, properties that make deployed agents harder, not easier, to abuse.

Environmental impact. Both datasets total under 18k training pairs and each fine-tune completes on one to eight GPUs; the full Dataset-B generation cost $\sim$\$220 of API inference. The method's data efficiency is itself a sustainability property.

REPRODUCIBILITY STATEMENT

Training configurations for every reported run (LoRA ranks, learning rates, β, sequence lengths, hardware, and checkpoint selection), the evaluation protocols P1–P3 with benchmark versions, serving stack, seeds, and the base-model sanity gate are given in Appendix A. Dataset composition, judge issue-label breakdowns, slice construction for the ablations, and degenerate-row filtering are described in Appendix B; the scenario library is described in Appendix L. Code, both datasets, the generation pipeline, and model weights will be released upon publication.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic datasets: the training data studied in this paper is produced by large language models acting as Scenario Generator, User Simulator, Teacher, and Judge, as specified in Section 3 and Appendix B; this is the method under study. We also used a coding agent to assist in implementing the training and evaluation code and in cleaning and reformatting data. We have not used generative AI tools for developing theoretical models or conceptual frameworks, formulating mathematical claims, writing proofs, proposing or refining hypotheses, designing research methodology, or interpreting results; translation and qualitative analysis support are not applicable to this work. Additionally, we used generative AI tools for creating figures and visualizations and for editing and polishing the manuscript text. We have reviewed all AI-assisted work: all AI-assisted code was reviewed and tested by the authors, and all edited text was checked by the authors against the underlying experimental results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi 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, 2024.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9354–9366, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.

Sungjin Jung, Dongjun Lee, Seungwon Lee, Gwanhyeong Seo, Dongha Lee, Bonhyeok Ko, Junhee Cho, Kyungmin Kim, Eunbi Kim, and Minsoo Shin. Diatool-dpo: Multi-turn direct preference optimization for tool-augmented large language models. *arXiv preprint arXiv:2504.02882*, 2025.

Jack Lanchantin, Angelica Chen, Janice Lan, Xian Li, Swarnadeep Saha, Tianlu Wang, Jing Xu, Ping Yu, Weizhe Yuan, Jason E Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Bridging offline and online reinforcement learning for LLMs. *arXiv preprint arXiv:2506.21495*, 2025.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. *arXiv preprint arXiv:2304.08244*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Yu Pan, Zhongze Cai, Guanting Chen, Huaiyang Zhong, and Chonghuan Wang. What matters in data for dpo? *arXiv preprint arXiv:2508.18312*, 2025.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 126544–126565, 2024.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 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 *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18:186345, 2024.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*, 2024.

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.

Xin Yu, Liuchen Liao, Yiwen Zhang, Yingchen Yu, Lingzhou Xue, and Qinzhen Guo. Preference-based self-distillation: Beyond KL matching via reward regularization. *arXiv preprint arXiv:2605.05040*, 2026.

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.

A REPRODUCIBILITY DETAILS

A.1 GENERATION-SIDE MODEL VERSIONS

Dataset A was generated with Fireworks-hosted `kimi-k2-thinking` (Teacher, Judge and Scenario Generator) and `qwen3-coder-480b-a35b-instruct` (User Simulator), with `Qwen/Qwen3-30B-A3B-Instruct-2507` as Student. Dataset B used OpenRouter-hosted `deepseek/deepseek-v4-pro` (Teacher), `z-ai/glm-5.2` (Judge and Scenario Generator) and `tencent/hy3-preview` (User Simulator), with `ibm-granite/granite-4.1-8b` as Student. Evaluation used OpenAI `gpt-4.1` as the τ^2 user simulator under P1 and P2, and `deepseek-v4-flash` via OpenRouter under P3. Dataset A was generated in December 2025 and Dataset B on 13–15 July 2026; evaluations ran in September 2026. These are provider-hosted endpoints without immutable version pins, so exact reproduction depends on provider snapshots; the pipeline itself is provider-agnostic and can be re-instantiated with open-weight models.

A.2 TRAINING CONFIGURATIONS

30B flagship: The retained configuration for the reported DPO continuation uses LoRA r=16/α=32, lr 2×10^{-6}, β=0.15, bf16, max seq 16,384, 8×H100, checkpoint-1410 (dev-selected). **4B/8B ablations:** LoRA r=16/α=32 on all seven projection matrices, lr 2×10^{-6} cosine (warmup 0.03), β=0.15, sigmoid loss, 3 epochs, effective batch 16 (bs 1 × grad-accum 16), gradient checkpointing, bf16, max seq 8,192 (Dataset A) / 10,240 (Dataset B, fitted to its longer conversations; constant across conditions). Dev split: 3% held out per slice, seed 42; final ≈ dev-best checkpoints for all reported conditions (verified from dev-loss trajectories).

A.3 EVALUATION DETAILS

P2 uses τ^2-bench with vLLM serving and per-family tool-call parsers (`hermes` for Qwen, `granite4` for Granite); GPT-4.1 user simulator; 3 seeds; full task sets (airline 50 / retail 114 / telecom 114) for every cell of Table 3, all re-measured under τ^2-bench v1.0.1 with a 16,384-token serving context. An earlier evaluation of the Dataset-A Granite cell used per-domain subsets and a different serving stack, and an earlier evaluation of the Dataset-A Qwen cell (base 0.303, distill 0.337, control 0.347, full 0.359), taken two months earlier under an earlier build of the same stack, is superseded by the re-measurement (0.291 / 0.305 / 0.338 / 0.352). Between the two independent measurements every condition moved by 1–3 points and the ordering base < distill < control < full held in both; the *full−distill* contrast read 2.2 then 4.7 points, a change inside that contrast's own interval (+0.6 to +8.7), so we read it as seed-to-seed variation on a noisy contrast rather than an effect of the stack. We report the current build because it is the one under which every cell in the paper was measured, with the parser check of Appendix A.4 applied to each model family. The Dataset-B evaluations ran τ^2-bench 1.0.1 with vLLM 0.25.1, trl 0.15.2, transformers 4.57.6 (training) / 5.14.1 (evaluation), and torch 2.11–2.12 (CUDA 13); earlier P2 runs used the same benchmark task definitions under earlier builds of the same serving stack. P3 uses per-domain train/held-out splits (airline 30/20, retail 68/46, telecom 68/46; split seed 42), DeepSeek-V4-Flash user simulator for all arms, 3 seeds on held-out tasks. GRPO-Env: 8 iterations, 4 sampled episodes per task per iteration, temperature 0.8 rollouts, group-relative advantages (zero-variance groups dropped), ratio-clipped policy gradient (ϵ=0.2) on assistant tokens, lr 1×10^{-5}, LoRA merged between iterations. Both GRPO arms use the same LoRA configuration as every DPO run (r=16, α=32, dropout 0.05, the same seven projection matrices), so the offline/online comparison differs only in algorithm and reward source. The GRPO-TSJ environment is closed-loop but stateless: an LLM user simulator role-plays each of the 1,926 deduplicated TSJ conversations from its persona and reacts to whatever the policy says; tool calls return the recorded response when the tool name matches the recording and an LLM-generated response otherwise; the reward is an order-aware, argument-graded match of executed tool calls against the Judge-chosen actions, with a penalty per reproduced forbidden action. Off-trajectory states are therefore handled by generation rather than by a transition model: there is no persistent database, so tool outputs after a novel action need not be consistent with earlier ones, and valid alternative action sequences earn no credit. Both properties bias this arm downward, which is why we report it as a lower bound.

Within-window subset (4B ablation). Using the per-turn token counts stored with each episode, 30% of the 4B/Dataset-A episodes (airline 59%, retail 35%, telecom 12%) never exceed 8,192 tokens, the 4B training length. On that subset the four conditions score base 0.468, distill 0.470, control 0.504, full 0.525 (all-episode values 0.287 / 0.300 / 0.333 / 0.345), and the task-level *full*−*base* contrast is $+0.167$ on airline (n=16 tasks, interval $+0.04$ to $+0.31$) and $+0.108$ macro. The subset is selected on episode length, which the policy influences, so this is descriptive; it shows the gain is not confined to episodes that ran past the training window. A direct 8,192-context evaluation was attempted and abandoned: 22% of airline episodes overflow the context mid-conversation and are unscorable. Script: `experiments/stats/within_window.py`.

Airline seed distribution (30B). To check whether three seeds understate airline, we ran ten further seeds (4–13) for the flagship and its base under P1 (agent temperature 0, GPT-4.1 user simulator). Flagship: 0.36, 0.40, 0.34, 0.26, 0.40, 0.28, 0.40, 0.34, 0.38, 0.42, 0.38, 0.32, 0.38 (mean 0.358, sd 0.049). Base: 0.34, 0.32, 0.34, 0.24, 0.34, 0.38, 0.28, 0.32, 0.40, 0.30, 0.36, 0.34, 0.32 (mean 0.329, sd 0.041). The flagship is higher in 9 of 13 seed-matched pairs; the task-level bootstrap on all 13 seeds gives $+0.029$ (-0.011 to $+0.072$). Under a different user simulator (`gpt-5.6-sol`, 3 seeds) the two models are likewise indistinguishable (0.340 vs. 0.360). Table 1 keeps the three-seed protocol shared by all domains.

Intervals. Every interval in this paper is a task-level bootstrap: within each domain, tasks are resampled with replacement (10,000 draws), each task contributing its mean reward over the 3 seeds; per-condition values and macro-averages are recomputed on each draw, and contrasts use the same resampled task set for both conditions (paired). Tables report the half-width of the 2.5–97.5% range as $\pm$; the text gives the range. This replaces the $1.96\,s/\sqrt{3}$ seed-level intervals of an earlier draft, which were roughly half as wide and treated three seed means as the sample; the task-level interval reflects task heterogeneity, the dominant source of variance here. It resamples tasks, not seeds: each task enters with its mean over the three seeds, so the interval represents uncertainty from task heterogeneity at the observed seed average, and does not capture seed-to-seed or run-to-run variance (user-simulator and serving nondeterminism), which it averages out. We therefore also report "positive in every seed" where it holds, and the 13-seed airline study above shows the seed dimension directly.

Excluded simulations. A simulation that ends in a harness infrastructure error carries no reward and is excluded from its seed's mean rather than scored as a failure. Across the 16 conditions of Tables 1 and 3 this affects 38 of 13,344 simulations (at most 9 of 834 in any condition, none in the trained Qwen arms); scoring them as zero instead changes no macro-average by more than 0.004 and no reported contrast's significance.

The SFT observation. An SFT run on the same 7,747 chosen trajectories, evaluated under an earlier protocol whose numbers are not comparable with Table 1 (airline 0.18, retail 0.16, BFCL 28.4%), drove training loss down while agentic scores collapsed. We report it as an observation about a configuration that mixes no general-instruction data, not as a baseline; the checkpoint was not retained, and no later claim depends on it.

External baselines. An earlier draft listed published τ-bench figures for xLAM-2-32b and Qwen2.5-72B next to our own. We removed them after re-measuring xLAM-2-32b on the P1 harness (τ^2-bench v1.0.1, `xlam` parser, 16,384-token context, 3 seeds, full task sets): airline 0.380 ± 0.117, retail 0.629 ± 0.072, telecom 0.208 ± 0.061, macro 0.405, against 0.379 for our 30B model and 0.315 for its base. Its published retail figure was 0.41. On a shared harness the specialist is significantly better on telecom ($+0.076$, interval $+0.026$ to $+0.129$) and indistinguishable on airline, retail and macro ($+0.027$, interval -0.016 to $+0.068$); we make no claim to outperform it; the paper's claims concern what TSJ pairs add to a given student, not a ranking of systems.

A.4 The parser cautionary tale

An initial cross-family attempt on Llama-3.1-8B produced near-zero retail/telecom scores *in every condition, including base*, with zero seed variance. The cause was a serving-stack tool-call parser bug that silently dropped all tool calls; the run was discarded and a base-model sanity gate (halt on near-zero domains, zero variance, or parser errors) was added to the harness. We report this because silent parser failures are indistinguishable from "the method does not generalize" unless base-model health is gated first.

B DATASET COMPOSITION

Dataset A (Qwen-native): 7,747 pairs from ∼1,926 conversations; 71% teacher-win / 29% judge-synthesized; generated with Kimi-k2-thinking (teacher, judge, scenarios) and Qwen3-Coder-480B (user simulator). The 2,257 judge-synthesized negatives are all free text (0% contain tool calls). Their multi-label issue counts (a negative can carry several tags, so these are not the single primary categories of Appendix K, where tone-only is 31%) are instruction violation 1,040; style 1,030 (45.6%); missing steps 887; factual error 540; fundamental error 386; safety 312; and tool misuse 176. Thus, Dataset A's synthesized-negative branch primarily represents textual failure descriptions, unlike Dataset B's rejected replies, which include executable tool calls.

Dataset B (Granite-native): 10,137 pairs from 1,333 conversations (mean 13.9 assistant turns; up to 24); 86% teacher-win / 14% judge-synthesized; teacher DeepSeek-V4-Pro, judge GLM-5.2, scenario generator GLM-5.2, user simulator Hunyuan-3-preview. Judge issue labels (multi-label, describing the losing reply), split by branch. Teacher-win rows (8,707 real student failures): factual error 7,287; instruction violation 2,688; tool misuse 2,452; missing steps 2,289; style 1,041; safety 356. The 1,430 judge-synthesized negatives: style 460 (32.2%); missing steps 394; instruction violation 371; factual error 263; tool misuse 186; safety 44. 2,054 chosen and 2,397 rejected replies contain executable tool calls (487 multi-tool); 139 rejected replies call hallucinated tool names; these are retained deliberately as genuine student failures. All 2,397 tool-call-bearing rejected replies come from the teacher-win branch (real student failures); the 1,430 judge-synthesized negatives are free-text assistant turns by construction, and their violations are behavioral (skipped verification, unverified commitments, premature promises) rather than malformed tool syntax. The same holds for Dataset A. System prompts: 52% voice-framed / 48% channel-neutral. Rows within a conversation share prefixes (strict-prefix snapshots per turn); conversation-level splits should deduplicate by scenario id. Dataset B will be released with per-pair provenance metadata (`answer_sources`, judge issue tags).

Slices (P2 ablations). Dataset A: full 7,747 / distill 5,490 / control 5,490 (seeded subset). Dataset B: full 10,108 / distill 8,681 / control 8,681, after dropping 29 degenerate rows (empty replies or judge-copied pairs).

C PIPELINE DIAGRAM

Figure 2 shows the generation loop described in Section 3.

D PASS^K ON THE FLAGSHIP

Table 4 gives the pass^k values summarized in Section 5.1.

Table 4: pass^k under τ^2-bench v1.0.1 (K=4 trials/task, GPT-4.1 user simulator). Retail additionally pools two 4-trial samples per model (K=8). These runs share the τ^2 release of protocol P1 but were served at a 32,768-token context rather than 16,384, so k=1 values differ slightly from Table 1; telecom in particular is dominated by the aborted-episode effect described above.

	Airline		**Retail**		**Telecom**	
k	base	TSJ-DPO	base	TSJ-DPO	base	TSJ-DPO
1	0.380	**0.405**	0.538	**0.630**	0.018	**0.151**
2	0.253	**0.287**	0.414	**0.484**	0.000	**0.056**
4	0.160	**0.220**	0.312	**0.352**	0.000	**0.009**
8	—	—	0.214	**0.263**	—	—

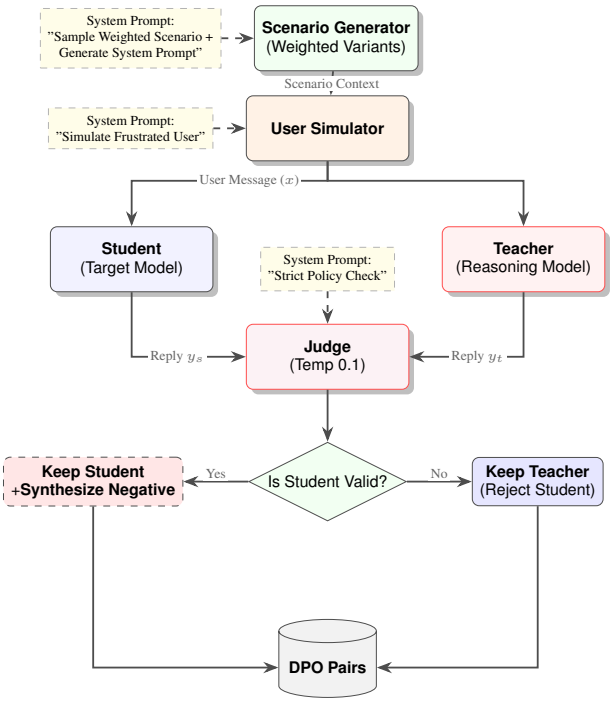

Figure 2: **The Teacher–Student–Judge pipeline.** A User Simulator generates adversarial multi-turn inputs from weighted scenarios. At every assistant turn the Student (the model to be trained) and the Teacher answer the same context; the Judge selects the winner between the Student's reply y_s and the Teacher's y_t under strict policy and schema checks. If the Student is valid, it is kept as y_w and the Judge *synthesizes* a free-text negative y_l (e.g., an unjustified refusal or an ungrounded claim that verification occurred); if invalid, the Teacher is kept and the Student's actual reply becomes y_l. The pipeline is model-agnostic: we instantiate it with two disjoint model ensembles (Table 8).

E Per-domain results for the student-matching experiment

Tables 5 and 6 give the per-domain values behind the four cells of Table 3.

Table 5: Dataset-B per-domain outcomes for Granite-4.1-8B. Values are means over three seeds ± the half-width of a task-level bootstrap 95% interval. Airline and telecom deltas compare full with base; retail compares control with distill.

Condition	Airline	Retail	Telecom	Macro average
Base	0.192 ± 0.096	0.567 ± 0.075	0.118 ± 0.044	0.292
Full	0.245 ± 0.103	0.585 ± 0.070	0.162 ± 0.056	0.331
Distill	0.241 ± 0.092	0.541 ± 0.073	0.129 ± 0.047	0.304
Control	0.294 ± 0.109	0.588 ± 0.070	0.163 ± 0.056	0.348
Reported contrast	**Full − base: +0.053**	**Control − distill: +0.047**	**Full − base: +0.044**	

The two mismatched cells fail in different ways; Section 8 discusses the contrast and why we do not claim a single mechanism for it.

F Offline vs. online at 4B

Table 7 gives the per-domain values behind Section 9.

Table 6: Per-domain values for the three remaining cells of Table 3 (τ^2-bench v1.0.1, 16,384-token context, full task sets, 3 seeds). The Dataset-B Granite cell is Table 5. Each family shares one base measurement across its two dataset rows.

Student / data	Condition	Airline	Retail	Telecom	Macro
Qwen3-4B / A (matched)	base	0.307	0.443	0.123	0.291
	distill	0.327	0.447	0.140	0.305
	control	0.360	0.482	0.173	0.338
	full	**0.380**	**0.518**	0.158	**0.352**
Qwen3-4B / B (mismatched)	base	0.307	0.443	0.123	0.291
	distill	0.300	0.485	0.175	0.320
	control	0.300	0.494	0.175	0.323
	full	0.280	0.447	0.164	0.297
Granite-4.1-8B / A (mismatched)	base	0.192	0.567	0.118	0.292
	distill	0.204	0.553	0.070	0.276
	control	0.184	0.506	0.097	0.262
	full	0.240	0.558	0.138	0.312

Table 7: Offline vs. online at 4B (protocol P3: held-out tasks, 3 seeds, shared user simulator). Mean $\pm$ half-width of a task-level bootstrap 95% interval; macro-averages use unrounded seed-level means.

Method	Airline	Retail	Telecom	Avg
base (Qwen3-4B)	0.583$\pm$0.167	0.391$\pm$0.116	0.239$\pm$0.109	0.405
GRPO-TSJ (synthetic env)	0.467$\pm$0.167	0.362$\pm$0.101	0.246$\pm$0.105	0.359
GRPO-Env (in τ^2)	**0.533$\pm$0.167**	0.406$\pm$0.112	**0.319$\pm$0.105**	0.419
TSJ-DPO (offline)	0.483$\pm$0.167	**0.500$\pm$0.116**	0.312$\pm$0.101	**0.432**

G TSJ INSTANTIATIONS

Table 8 lists the models filling each role in the two pipeline runs of Section 3.1; dataset statistics are in Appendix B.

Table 8: TSJ instantiations. Every role is filled by a different model across the two datasets, guarding against ensemble-specific effects; only the pipeline is shared.

Role	Dataset A (Qwen-native)	Dataset B (Granite-native)
Student	Qwen3-30B-A3B-Instruct-2507	Granite-4.1-8B
Teacher	Kimi-k2-thinking	DeepSeek-V4-Pro
Judge	Kimi-k2-thinking	GLM-5.2
Scenario gen.	Kimi-k2-thinking	GLM-5.2
User simulator	Qwen3-Coder-480B	Hunyuan-3 (preview)
Pairs	7,747	10,137
Conversations	$\sim$1,926	1,333
Teacher-win / synthesized	71% / 29%	86% / 14%

H LENGTH AND TONE CONTROLS FOR THE STYLE-SHORTCUT ARGUMENT

Scope of the argument. A one-dimensional style heuristic is *strictly* self-contradictory only when Teacher and Judge are the same model, as in Dataset A (both Kimi-k2-thinking); in Dataset B the three authors are distinct, so a transitive multi-class style ranking remains learnable in principle. Figure 1 shows the shortcut is suppressed there empirically, so we present the mechanism as the best-supported account, not a proof. Structural differences between families (tool-call syntax, turn length, tag conventions) are part of what "failure surface" means here; the mismatched cells cannot separate a style-surface mismatch from a policy-surface one.

Length bias is not the shortcut. A shallow feature that separates y_w from y_l could simply be response length. It is not: in the *distill* pairs, the chosen reply is longer than the rejected one in 49% of Dataset-A rows (mean 177.9 vs. 178.0 tokens; median 168 vs. 158) and only 32% of Dataset-B rows (219 vs. 304), so a length heuristic would be near-chance or actively wrong on the very pairs that reach near-zero loss. In the synthesized pairs the judge-written negative *is* shorter (median 41 and 66 tokens vs. 133 and 95 for the student's reply); that cue points the opposite way to the Dataset-B distill rows and is absent in Dataset A, so it cannot account for the transfer gain either. Length-matched negative synthesis is nonetheless a sensible refinement.

Tone is not the shortcut either. Our audit finds 31% of synthesized negatives differ mainly in tone (Appendix K) — apparently the cue these rows should remove. The dynamics argue against it: an easy tone discriminator would make the held-out task *easier*, yet adding these rows raises dev loss an order of magnitude. Our interpretation is that such rows dilute the slice rather than drive it. This is an inference from loss curves, not a controlled test; a tone-filtered or length-matched training ablation would settle it and was not run.

I COMPOSITION LIMITATIONS OF THE TRAINING DATA

Are the negatives "hard"? We scored every Dataset-A pair under the untrained Qwen3-4B (the P2 student), rendering prompt, chosen and rejected exactly as in training and taking the mean per-token log-likelihood of each reply given the prompt. In the 5,490 teacher-win pairs the base policy prefers its *own* rejected reply in 75% of cases (mean per-token margin $\log \pi_0(y_w) - \log \pi_0(y_l) = -0.32$ nats): those negatives are on-policy and the teacher's positive is not. In the 2,257 Judge-synthesized pairs the base policy prefers the chosen reply in 97% of cases (margin $+2.64$ nats; the judge-written negative averages -3.96 nats per token against -1.32 for the student's own reply). By a likelihood-margin definition the synthesized negatives are therefore *easy*, not hard, and the paper's claim is about negatives anchored to the student: what the synthesized branch adds to the data is an on-policy y_w paired with an off-policy y_l, the mirror image of the teacher-win branch. That both branches are needed (Section 6) and that the on-policy side must belong to the deployment model (Section 7) is the paper's claim; that the negatives sit near the policy's decision boundary is not, and the measurement rules it out for this Judge. Script: `experiments/trial7_ctx8k/hardness.py`.

Synthesized negatives carry no tool syntax. By construction the Judge writes free text, so *zero* synthesized negatives in either dataset contain tool calls, while 46.7% (Dataset A) and 27.5% (Dataset B) of teacher-win negatives do. Every rejected sample that demonstrates malformed or policy-violating *syntax* therefore sits in exactly the slice that Section 8 shows is solvable by the style shortcut. That is a plausible mechanism for the pattern in Table 1: large gains on behavioral policy adherence but a modest BFCL move ($34.11\% \rightarrow 37.16\%$). Synthesizing negatives with deliberately incorrect tool calls is the most direct extension.

Exposure bias in the training prefixes. Since the winning reply is appended before the next turn and the Teacher wins 71–86% of turns, most prefixes are Teacher-written, whereas at deployment the model navigates its own history. The synthesized slice partially offsets this (those rows condition on the Student's accepted reply), but the imbalance argues for raising the synthesized fraction. That fraction is itself student-dependent, set by how often the Student wins (29% for the 30B, 14% for Granite-8B), so the mechanism's headroom shrinks as students weaken; a synth-fraction sweep is the clearest next experiment.

J ABORTED-EPISODE COUNTS

Table 9 gives the per-seed counts behind the behavioral signature of Section 7.

Table 9: Telecom episodes terminated as `too_many_errors` (of 114), per seed, Granite-4.1-8B. *control* and *full* contain synthesized negatives; *distill* does not.

Condition	Dataset B (matched)	Dataset A (mismatched)
base	60 / 70 / 70	60 / 70 / 70
distill	26 / 36 / 30	78 / 68 / 74
control	**2 / 1 / 6**	63 / 78 / 70
full	**3 / 1 / 1**	48 / 48 / 46

K AUDIT OF THE SYNTHESIZED NEGATIVES

How the negatives are produced. Both datasets use the same Judge system prompt (Dataset B adds one rule restricting the `safety_violation` tag to harmful content, so that procedural tool errors are tagged `tool_misuse`). The Judge sees the conversation history, the tool definitions and both candidate replies, returns a winner, and tags the losing reply with labels from a fixed set (instruction violation, factual error, tool misuse, missing steps, safety violation, fundamental error, style issue). It may declare the Teacher the winner only when that reply is *materially* better; ties go to the Student. The synthesis rule reads, verbatim: "When `winner = "student"`, also synthesize a short 'rejected' assistant reply (2–4 sentences) that clearly demonstrates the failure mode captured in `issues`. This string must reflect what a bad assistant might say for the same user request (e.g., unjustified refusals, laziness, hallucinated facts, unsafe guidance) [. . .] Do not copy the student's response." Two further rules forbid rewriting either input reply and fabricating validation tokens or bypassing safety gates described in the history. The negative is therefore a freshly written failure at a turn the Student handled correctly, not an edit of the Student's reply, and the 2–4-sentence constraint explains three properties reported below and in Section 10: the negatives are short, contain no tool-call syntax, and are often distinguished by tone.

Audit. We sampled 100 Judge-synthesized negatives from Dataset A (seed 7, fixed before reading) and one author read each against its conversation context and the Student's accepted reply, assigning one primary category from the four fixed in advance. The reader saw which reply was the synthesized negative (unavoidable, since the negative is by construction the Judge's text) and the Judge's issue tags. Artifact rates were then counted automatically over all 2,257 (Dataset A) and 1,430 (Dataset B) synthesized negatives.

Table 10: Categorization of 100 sampled Judge-synthesized negatives (Dataset A).

Category	n	Example
Policy / behavioral violation	47	acts before verifying identity; invents a ticket number
Unjustified refusal	17	"I cannot help with returns today"
Tone or terseness only	31	"Order updated. Goodbye."
Formatting artifact	5	literal `<tool_calls>` JSON inside the text

Counting artifacts anywhere in the text rather than as the primary category gives 7/100 for Dataset A. Narrow automatic regexes over the full sets find pseudo tool-call syntax in 0.7% (A) and 0.1% (B), replies under 15 words in 12.8% and 5.0%, and explicit refusals in 9.3% and 0.2%; these are lower bounds relative to the manual pass. Dataset B, whose Judge is GLM-5.2 rather than Kimi-k2-thinking, is markedly cleaner on terseness and refusals, so negative quality depends on the Judge as well as on the Student. Two cheap improvements follow: reject synthesized negatives containing tool-call-shaped strings, and instruct the Judge to match the accepted reply's length and register so that tone cannot separate the pair.

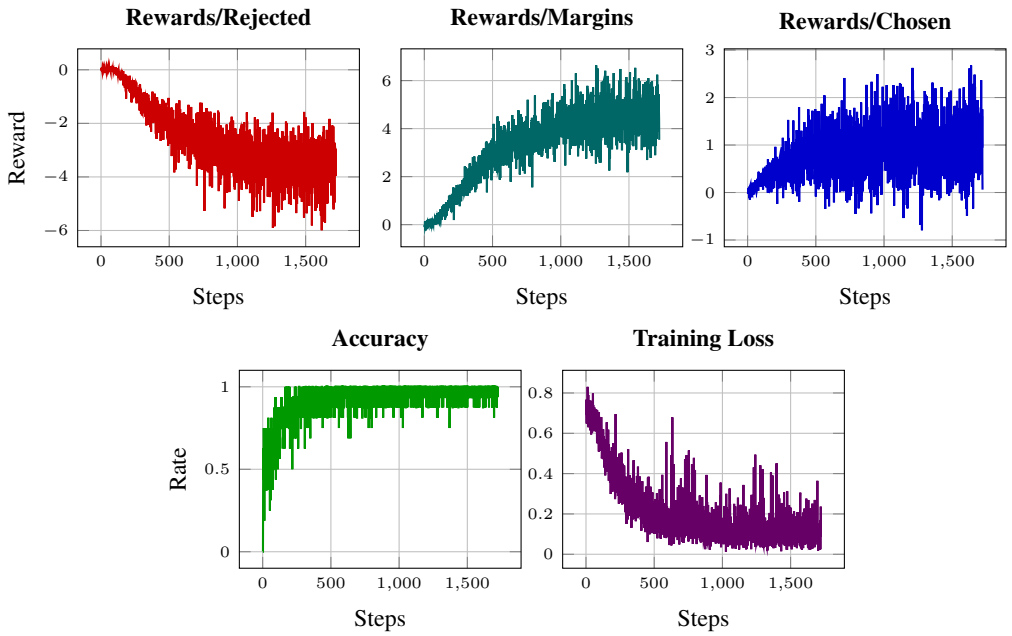

Figure 3: **Flagship DPO training dynamics (30B).** The model maximizes margins primarily by pushing down rejected-trajectory likelihood ($0 \rightarrow\approx -3.5$) rather than raising chosen likelihood ($\rightarrow\approx 1$): the objective is dominated by *unlearning* reachable failure modes, consistent with the negative-side attribution of Section 6. Curves cover the full 1,722-step run; the reported model is checkpoint-1410, selected on dev loss, so training continues past the selected point.

L SCENARIO LIBRARY

The 34 weighted variants and their sampling weights are listed below; the four-bucket grouping of Section 3 is post-hoc reporting only. Weights are relative and normalized at sampling time. Each variant additionally carries a requirement list (the behaviors the scenario must exercise) in the released pipeline configuration.

Table 11: The 34 scenario variants with sampling weights (share of generated scenarios).

Variant	%	Variant	%	Variant	%
single_tool_parameter_precision	5.9	multi_turn_long_context_filtering	3.7	airline_delay_comp_certificate_pivot	0.9
multi_tool_routing_with_decoys	5.9	multi_turn_composite_challenge	3.7	misinformation_detection	0.7
parallel_independent_tool_calls	5.9	web_search_snippet_grounding	3.7	false_premise_challenge	0.7
multi_turn_contextual_resolution	5.9	web_search_fullpage_reasoning	3.7	knowledge_limits	0.7
parallel_mixed_routing	5.2	memory_vector_store_management	3.7	unsafe_advice	0.7
multi_turn_missing_parameters_blocker	5.2	memory_key_value_discipline	3.0	retail_cancel_confirmation_conflict	0.7
complex_task_resolution	4.4	memory_recursive_summarization	3.0	airline_passenger_baggage_edits	0.7
irrelevant_tool_refusal	4.4	rest_and_sql_precision_calls	3.0		
format_sensitive_output_contract	3.7	typed_language_tooling_accuracy	3.0		
multi_turn_missing_function_escalation	3.7	tau_gap_coverage	2.6		
multi_domain_branching_workflow	2.6	tau_canonical_retail_tools	1.8		
edge_case_tool_conflict_resolution	1.5	tau_canonical_airline_tools	1.8		
retail_multi_modify_fallback	1.5	retail_return_exchange_multi_order	1.3		
airline_nonstop_change_downgrade	1.1				

M TRAINING DYNAMICS OF THE FLAGSHIP RUN

Negative constraint learning. We use this term for the training-dynamics perspective on the same student-matched negative mechanism studied in Sections 6–8. As Figure 3 shows, `rewards/rejected` falls to ≈ -3.5 while `rewards/chosen` rises only to ≈ 1: DPO here

operates mainly by carving out the negative space of the policy. This is the flagship-scale echo of Section 8: the negatives do the work, which is why their provenance matters so much.

Delayed behavioral convergence. Benchmark trajectories across checkpoints (Figure 4) show simple tool workflows (Retail) mastered early while the strict Airline policies converge late, with a transient regression near step 1200 before the jump to 0.44 at step 1410. Pairwise *discrimination* accuracy climbs quickly (above 0.9 within the first 300 steps, approaching 1.0 well before the midpoint of training), revealing a **discrimination–generation gap**: ranking the correct response is learned long before reliably *generating* it over multiple turns. We use "delayed behavioral convergence" descriptively and make no phase-transition claim; the single-checkpoint 0.44 reading should be interpreted with its ± 0.02 CI.

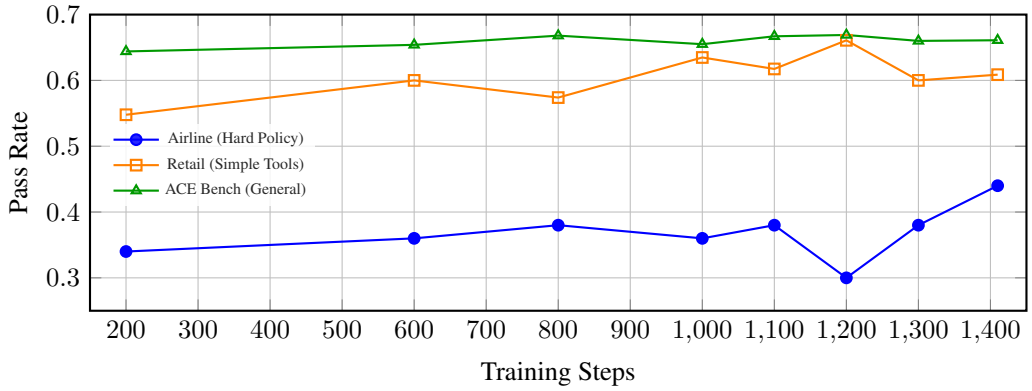

Figure 4: **Delayed behavioral convergence (30B).** Retail and ACE improve steadily; Airline stays in the mid-30% range with a transient regression near step 1200, consistent with late consolidation of strict verification policies. These checkpoint evaluations were run during training under the original protocol; absolute values are not comparable with Table 1, which re-measures the selected checkpoint under τ^2-bench v1.0.1.

N QUALITATIVE ANALYSIS

A signature failure of base models is *premature tool execution*: acting before verifying. The trained model instead follows verify-then-act workflows (Table 12), the same behavior that the aborted-episode counts of Appendix J measure in aggregate, and precisely what the synthesized negatives penalize.

Table 12: **Task 6 comparison (flight modification, 30B).** The base model acts blindly; ours follows a verify-then-act workflow.

Base Model (Fail)	**Ours (Success)**
User: Change to the cheapest flight tomorrow.	*User: Change to the cheapest flight tomorrow.*
Step 1 (Immediate Action):	**Step 1 (Info Gathering):**
`update_reservation( flights=["HAT110"]... )`	`search_onestop_flight( date="2024-05-24"... )`
Outcome: FAILURE	**Outcome:** SUCCESS
Hallucinated that HAT110 was cheapest without searching; API rejected the booking.	Found 5 options, identified the cheapest ($206), confirmed with user, *then* booked.

Remaining failure modes. Of the Airline tasks the 30B model still fails, roughly 40% are multi-hop reasoning decay (the verification chain is initiated but lost after 3+ turns of tool output), 30% ambiguity resolution (guessing a parameter instead of asking), and the remainder rare tool combinations under-

represented in the synthetic distribution. The model *ranks* correct policies confidently (high reward margins) yet drifts during long-horizon generation: the discrimination–generation gap again.