# REPLAY-VERIFIED FACTOR-LEVEL DISTILLATION FOR TOOL-TRACE REPAIR

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Distilling tool use from a black-box teacher is complicated by a simple failure mode: a repair can be fluent and schema-valid yet remain incorrect under execution. We introduce replay-verified factor-level distillation (RVFD), a procedure that treats teacher outputs as proposals rather than ground truth. RVFD decomposes a proposed repair into schema-aligned factors, applies each edit to the failed trace, and retains an edit as supervision only when replay reaches the desired outcome. On BFCL-CF, full-parameter Qwen2.5-7B and 14B students trained with RVFD improve greedy repair success over direct teacher-output SFT by $0.0547$ and $0.0534$, respectively, without a teacher or verifier at test time. An augmented factor student reaches $0.9199$ greedy success, compared with $0.8893$ for whole-trace execution-filtered SFT; this comparison also differs in training-set size. On naturally occurring errors in archived BFCL outputs, local repair raises AST accuracy from $0.5939$ to $0.6774$, while whole-trace rewriting with beam4 and checker access reaches $0.8736 \pm 0.0059$. Optional test-time replay improves accuracy further, but does not surpass CAR-point or oracle-step CausalFlow on a shared candidate pool. These results support replay as a source of local credit assignment for tool-trace repair, rather than as a general solution to end-to-end agent learning. Code and data are available at https://anonymous.4open.science/r/rvfd-aaai27/.

## 1 INTRODUCTION

Knowledge distillation commonly assumes that outputs from a stronger model are useful training targets. This assumption is fragile for tool use. A tool trace is both a token sequence and an executable program, and surface plausibility does not imply behavioral correctness. A teacher may therefore produce a well-formed repair that leaves the original failure unchanged.

Consider a reservation update that uses `city` when the schema requires `location`. Renaming the key can fix the trace; changing the city value or inserting another call cannot. A teacher may suggest all three. Standard SFT gives each proposal the status of a training target, even though only one changes the outcome (Figure 1).

We study a local credit-assignment problem: which part of a teacher-proposed repair should supervise the student? We retain an edit only if applying it in isolation changes the failed trace into one accepted by the task checker. The teacher supplies plausible candidates, and verification determines the target. Whole-trace imitation instead supervises every choice in the corrected trace, including unchanged arguments and formatting unrelated to the repair.

We instantiate this idea as replay-verified factor-level distillation (RVFD). The method decomposes a repair into four schema-aligned factors: function, argument key, argument value, and stop/missing call. A black-box teacher proposes a local edit, which is then applied and re-executed; only successful interventions enter the training set. We distinguish two evaluation regimes throughout. Greedy decoding from the student checkpoint isolates the effect of label construction, whereas beam-plus-replay spends an explicit test-time verifier budget.

Experiments on BFCL-CF, a controlled repair benchmark derived from BFCL (Gorilla Team, 2024), support this distinction. Replay-validated factor targets improve full-parameter 7B and 14B students over direct teacher SFT and response-level filtering. A larger, augmented factor dataset also produces a stronger student than the whole-trace execution-filtered baseline, although that comparison does



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.

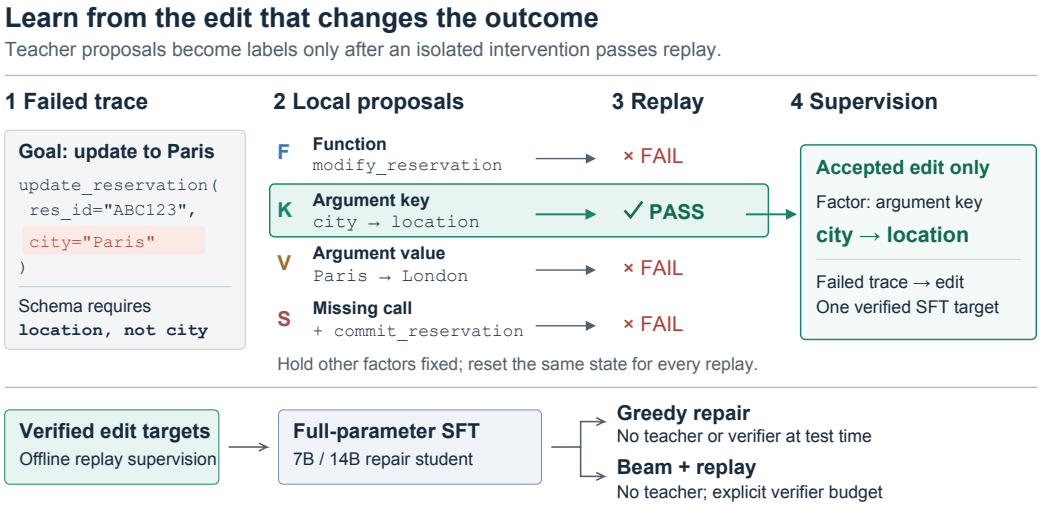



Figure 1: RVFD separates proposal, judgment, and learning. A black-box teacher proposes local edits to a failed trace. Each edit is replayed as an isolated intervention; only a successful factor becomes a training target. The main evaluation uses the student checkpoint alone. Beam-plus-replay is optional test-time verifier compute and is reported separately.

not isolate supervision granularity. Replay selection yields an additional gain when verifier compute is available, although CAR-point and oracle-step CausalFlow reach the same ceiling on an identical candidate pool. The empirical gain thus comes from constructing more informative training targets and student candidates, rather than from a new selection rule.

Our main contributions are:

- We formulate local tool-trace repair as factor-level counterfactual label construction, using a teacher for proposals and executable replay for acceptance.
- We construct BFCL-CF, a factor-balanced BFCL repair benchmark, and separate train-time verification from test-time replay in every comparison.
- We show that replay-validated factor targets improve greedy 7B and 14B students without test-time assistance, and report a separate augmented-data comparison against whole-trace execution-filtered fine-tuning.
- We evaluate the method against CAR, CausalFlow, DPO, cross-provider teachers, naturally occurring BFCL failures, and stateful tau2 tasks, clarifying both the empirical benefit and its limits.

## 2 Problem and Learning Object

A tool-use trace is $\tau = (x, a_1, o_1, \ldots, a_T, o_T)$, where $x$ is the user request, $a_t$ is either a tool call or a stop action, and $o_t$ is the tool observation. A checker $R(\tau) \in \{0, 1\}$ returns whether the trace reaches the target outcome. We focus on failures that admit a local correction to one tool interaction.

**Repair factors.** A candidate repair is identified by the schema component it touches: function, argument key, argument value, or stop/missing-call. The partition is operational: each factor corresponds to a concrete edit on a JSON-like tool call, and the resulting trace can be fed directly to the original environment checker.

**Replay-validated factor-level counterfactual label.** Given a failed trace $\tau$, a factor $z$, and teacher edit $e$, let $\mathrm{Apply}(\tau, e)$ denote the trace obtained by applying only that edit. The tuple $(\tau, z, e)$ is a replay-validated factor-level counterfactual label when

$$R(\tau) = 0, \qquad R(\mathrm{Apply}(\tau, e)) = 1.$$

The label is *replay-validated* because execution determines acceptance, *factor-level* because it contains a single local schema edit, and *counterfactual* because the edit is evaluated as an intervention on the failed trace rather than by similarity to teacher text. Sufficiency is relative to the restored state and checker: one successful intervention does not establish that the edit is necessary, uniquely correct, or globally minimal.

**False adoption.** A false adoption occurs when a selector picks a candidate whose replay does not succeed. We report this metric alongside repair success for every candidate-selection experiment. The metric concerns verifier acceptance; greedy student evaluation instead reports repair success, and can still emit incorrect repairs.

**Budget taxonomy.** Four kinds of computation matter. The *train-time teacher budget* generates proposals; the *train-time replay budget* decides which become labels. At deployment, the *decoding budget* determines how many student candidates are drawn, and an optional *verifier budget* replays them and picks the first success. Our main comparison spends only the first two; test-time replay is always reported separately.

## 3 METHOD

RVFD has three stages: proposal, validation, and learning. A black-box teacher proposes local edits, but nothing is treated as a label until it survives replay. Algorithm 1 gives the procedure.

**Counterfactual validation.** Response-level filtering asks a coarse question: keep this entire teacher output or not? RVFD asks a sharper one about a single intervention: would *this* edit, applied alone, make the failed trace succeed? The difference matters whenever a teacher response bundles a correct fix with incidental or conflicting changes. Replay isolates the sufficient intervention and discards edits that are merely plausible.

**Label construction.** The accepted data are

$$\mathcal{D} = \{(\tau_i, z_i, e_i) : R(\mathrm{Apply}(\tau_i, e_i)) = 1\}.$$

Unsuccessful proposals are discarded; we do not soften them into partial-credit targets. The student predicts the accepted edit in a fixed schema (operation, step index, factor, old value, new value), which puts all four repair types into one output space and is compact enough to verify exactly.

For context $c_i$ containing the request, tool schema, and failed trace, we optimize the usual token-level SFT objective on accepted edit tokens:

$$\mathcal{L}(\theta) = -\sum_{(c_i,e_i)\in\mathcal{D}} \sum_{t=1}^{|e_i|} \log p_\theta(e_{i,t} \mid c_i, e_{i,<t}). \tag{1}$$

The student learns to output the edit type and location together with its replacement value. The contribution is the construction of its targets; the optimizer remains standard supervised fine-tuning.

**Algorithm 1** RVFD label construction

**Require:** failed trace set $\mathcal{F}$, factors $\mathcal{Z}$, teacher $T$, checker $R$

$\mathcal{D} \leftarrow \emptyset$
**for** $\tau \in \mathcal{F}$ **do**
  **for** $z \in \mathcal{Z}(\tau)$ **do**
    $e \leftarrow T(\tau, z)$
    $\hat{\tau} \leftarrow \mathrm{Apply}(\tau, e)$
    **if** $e$ is schema-valid and $R(\hat{\tau}) = 1$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\tau, z, e)\}$
    **end if**
  **end for**
**end for**
**return** $\mathcal{D}$

**Inference regimes.** We evaluate two main inference settings. First, the student emits a single repair greedily with no teacher and no verifier. Second, it draws several candidates and replay picks the first success; this still needs no teacher but spends test-time execution. Checker-gated overlays and structured fallbacks change the deployment interface and are therefore reported separately.

**What a successful intervention establishes.** Verification ties the label to a particular input and task criterion. A key rename that succeeds on one trace need not be useful on another, and a failed single-factor edit may be part of a valid multi-factor repair. We therefore interpret accepted examples as sufficient local corrections, not as exhaustive explanations of failure. Nor does acceptance rank every alternative edit: two different repairs can both satisfy the checker. In the stateful setting, the same pre-intervention state must be restored before each candidate is tested; otherwise a previous candidate can change what the next replay measures. This requirement is automatic for the immutable BFCL-CF trace checker, but is a substantive assumption for interactive environments.

**Why factor granularity matters.** Whole-trace targets conflate the effective repair with every other choice made by the teacher. A trace can contain the correct value but the wrong stopping decision, or the correct function alongside a stale argument. A factor target supervises only the minimal intervention that changes the outcome. Conversely, a failed replay does not imply that the entire teacher response is poor; it shows only that the proposed edit is not sufficient on its own.

## 4 Experimental Setup

**BFCL-CF construction.** BFCL-CF is a controlled repair benchmark we build from BFCL (Gorilla Team, 2024). We start from BFCL v4 prompts and reference answers that canonicalize cleanly into JSON tool calls. The resulting diagnostic pool holds 11,773 factor-level corruptions: 3276 argument-key, 3276 argument-value, 1919 function, and 3302 stop-call records. Function errors come from substituting another in-schema API, key errors from renaming an argument, value errors from deterministic incorrect values, and stop-call errors from deleting a required action. A repair counts as successful only if the edited canonical trace matches the reference trace; unparseable, identity, and non-improving edits are rejected. Here replay means applying an edit and rerunning this deterministic checker, not executing a live API. Natural BFCL uses the official AST checker; tau2 uses a stateful environment. The distinction limits how broadly execution claims can transfer.

The main DeepSeek-chat proposal set has 1024 balanced records, of which replay accepts 851. We train on 681 of the accepted examples and hold out 170 for development. The accepted set breaks down into 254 argument-key, 209 argument-value, 252 function, and 136 stop-call repairs. Evaluation runs on a proposal-disjoint 512-example holdout balanced across the four factors, and the cross-provider experiment adds a source-disjoint split on top.

**Models and teachers.** The main students are full-parameter Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct (Qwen Team, 2024), trained with FSDP full-shard auto-wrap on Qwen2DecoderLayer, AdamW, learning rate $10^{-5}$, bf16, and three epochs. Each configuration is trained with three random seeds (17, 23, 31); we report mean $\pm$ standard deviation unless otherwise noted. Teacher/proposal sources are DeepSeek-chat for the main proposals, GPT-5.4 for stop-call augmentation, and GPT-5.4 plus Claude Sonnet 4.6 for the cross-provider study (DeepSeek-AI, 2024). API teachers use a fixed JSON repair schema, temperature 0.0, and max tokens 256.

**Baselines.** On the training side we compare against direct SFT on teacher JSON, SFT on response-level accepted traces, whole-trace execution-filtered SFT in the RFT/STaR family (Zelikman et al., 2022; Touvron et al., 2023), and a DPO-style preference objective (Rafailov et al., 2023). For selection we include first-candidate selection, learned rerankers, CAR-CI, CAR-point, and a CausalFlow adapter built on the released code (Shah, 2026; Bonagiri et al., 2026). We run CAR from its official implementation. For CausalFlow we retain the public trace and graph abstractions while fixing the candidate pool, which isolates selection from proposal generation. OPD, TCOD, and GKD appear only as same-surface tau2 stress variants, not as exact reproductions (OPD Contributors, 2025; Wang et al., 2026; Agarwal et al., 2024).

| Block | Teacher | Replay | Role |
|---|---|---|---|
| Greedy student | no | no | label quality |
| Beam selection | no | yes | verifier budget |
| Official overlay | no | checker | diagnostic |
| tau2 fallback | no | replay+fb | diagnostic |

Table 1: Evaluation contract. Every result specifies decoding budget, test-time teacher access, and verifier/replay access. The first block isolates train-time label construction.

| Setting | Student | Training target | Success | Paired evidence |
|---|---|---|---|---|
| Balanced | 7B | Direct teacher JSON | $0.7656 \pm 0.0078$ | – |
| Balanced | 7B | Response-filtered trace | $0.7799 \pm 0.0114$ | – |
| Balanced | 7B | **Replay-validated factor** | $\mathbf{0.8203 \pm 0.0070}$ | $+0.0547$, $p_{17} = 6.60\times10^{-3}$ |
| Balanced | 14B | Direct teacher JSON | $0.7663 \pm 0.0079$ | – |
| Balanced | 14B | Response-filtered trace | $0.7754 \pm 0.0034$ | – |
| Balanced | 14B | **Replay-validated factor** | $\mathbf{0.8197 \pm 0.0159}$ | $+0.0534$, $p_{17} = 2.09\times10^{-5}$ |
| Augmented | 7B | Whole-trace RFT/STaR | $0.8893 \pm 0.0088$ | – |
| Augmented | 7B | **Replay-validated factor** | $\mathbf{0.9199 \pm 0.0052}$ | $+0.0306$, all seeds $p < 0.036$ |

Table 2: Train-time label construction on BFCL-CF. All rows use greedy decoding with no test-time teacher or verifier. Balanced rows use the 1024-proposal study; augmented rows compare training recipes with different dataset sizes (1,023 factor vs. 681 trace rows). Mean $\pm$ std, seeds 17/23/31.

## 5 RESULTS

We organize the results around two questions. First, does replay improve the supervision used to train a standalone student? Second, what additional benefit comes from test-time verification? Table 1 separates these regimes. Cross-provider, official-checker, and tau2 experiments then probe the limits of transfer.

**Factor-level targets improve standalone students.** Table 2 reports greedy evaluation of full-parameter students with no teacher, verifier, or replay at test time. In the balanced setting, replay-validated factor targets improve 7B and 14B success by $5.47$ and $5.34$ points over direct SFT on teacher outputs; response-level filtering helps only marginally. The augmented comparison yields a further $3.06$-point gap over whole-trace RFT/STaR-style SFT. Its public training files contain 1,023 factor-target rows and 681 whole-trace rows, so the gap reflects the complete training recipes rather than a controlled estimate of target granularity.

**Test-time replay trades execution for accuracy.** Without a verifier the augmented factor student reaches $0.9199 \pm 0.0052$. Drawing several candidates and replaying them lifts success to $0.9583 \pm 0.0049$ at beam4 and $0.9733 \pm 0.0060$ at beam16, with zero false adoptions. This requires $3.75$ and $12.70$ replay calls per example on average, so the gain comes from test-time compute rather than additional training. Most residual failures sit in stop/missing-call repairs ($0.9297$ even at beam16); function, key, and value repairs are near saturation.

A DPO control reaches $0.6050$ greedy and $0.7050$ with beam4 plus replay. Replay selection again keeps false adoption at zero, but the large gap to replay-filtered SFT shows that the preference control underperforms exact-edit supervision in this configuration. This does not establish a general limitation of preference optimization.

**Published causal methods locate the selection ceiling.** Table 4 fixes the student candidate pool, replay budget, and seeds. RVFD outperforms the conservative CAR-CI rule and the first-intervention CausalFlow readout. CAR-point and oracle-step CausalFlow, however, match RVFD exactly. Once every candidate can be executed, immediate replay success already supplies an optimal selection rule. The relevant difference is upstream: factor-level supervision makes a successful repair more likely to occur among the student's candidates, even before selection.

| Decode | Success | Avg. replay calls |
|---|---|---|
| Greedy (no verifier) | $0.9199 \pm 0.0052$ | 0.00 |
| Beam4 + replay | $0.9583 \pm 0.0049$ | 3.75 |
| Beam16 + replay | $\mathbf{0.9733 \pm 0.0060}$ | 12.70 |

Table 3: Cost-quality curve for the augmented 7B factor student, seeds 17/23/31. No test-time teacher. Replay-selected candidates have zero observed false adoption under the same checker used for acceptance.

| Selector | Beam4 | Beam16 |
|---|---|---|
| CAR-CI | $0.2858 \pm 0.0176$ | $0.6829 \pm 0.0637$ |
| CausalFlow-first | $0.9258 \pm 0.0020$ | $0.9238 \pm 0.0000$ |
| CAR-point | $0.9583 \pm 0.0049$ | $0.9733 \pm 0.0060$ |
| CausalFlow-oracle | $0.9583 \pm 0.0049$ | $0.9733 \pm 0.0060$ |
| **RVFD replay** | $\mathbf{0.9583 \pm 0.0049}$ | $\mathbf{0.9733 \pm 0.0060}$ |

Table 4: Selector comparison on fixed 7B BFCL-CF candidate pools, seeds 17/23/31. All rows use executable replay with no test-time teacher; observed false adoption is zero throughout. The bottom three selectors hit the same ceiling.

**Repair transfers to naturally occurring BFCL failures.** Because BFCL-CF controls the corruption factor, we also test on 1,150 archived AST outputs from official BFCL v4 non-live evaluation: failures produced by a model, not by our corruption rules. The repair system fires only when the Gorilla AST checker rejects the original output; anything the checker accepts passes through untouched. Table 5 shows greedy rewriting fixing $46.61\%$ of original failures and beam4 fixing $68.88\%$, with no observed damage to already-correct outputs. This is a checker-gated overlay rather than a raw leaderboard submission, but it provides a distribution shift that owes nothing to our corruption process.

The natural evaluation also reveals where factor-local repair is insufficient. Factor prediction works best when the causal locus is known, as in Table 2. Naturally occurring outputs may contain several interacting errors simultaneously; for those, the whole-trace repair head has broader reach. Factor targets therefore provide a high-precision inductive bias for local repair rather than a replacement for trace-level rewriting.

**Transfer varies across tool-call structures.** The aggregate gain is not tied to one language or call pattern (Table 6). Greedy rewriting improves all six BFCL categories, including parallel and multi-call cases where errors can interact; beam4 pushes every category further. Java and JavaScript remain the hardest subsets, consistent with their wider variation in literal syntax relative to the Python-heavy training data.

**Accepted supervision transfers across providers.** Replacing DeepSeek with GPT-5.4 and Claude Sonnet 4.6 yields 473 accepted repairs out of 512 schema-completed proposals. A 7B student trained only on these rows reaches $0.8594 \pm 0.0034$ greedy and $0.9323 \pm 0.0069$ with beam4 replay on a source-disjoint holdout, confirming that the replay acceptance criterion is not specific to any one proposal provider.

**Free-form tau2 reveals a different bottleneck.** The 56-task tau2 evaluation pulls local repair apart from autonomous policy transfer (Table 7). CAR, OPD, TCOD, and the GKD control never clear $0.0893$ on this surface; a domain-routed student reaches $0.5536$. Even an oracle picking from all collected non-API candidates tops out at $0.6429$, meaning the candidate pool itself lacks solutions for many tasks. A hybrid with replay and deterministic fallback reaches $0.7500 \pm 0.0357$, but that is a repair-executor result with explicit assistance, not evidence of an equally strong standalone policy.

| Repair policy | Decode | AST accuracy | Rescue |
|---|---|---|---|
| Base official output | – | 0.5939 | – |
| Factor local repair | greedy | 0.6774 | 0.2056 |
| Whole-trace RFT/STaR | greedy | $0.7832 \pm 0.0033$ | $0.4661 \pm 0.0081$ |
| Whole-trace RFT/STaR | beam4 | $\mathbf{0.8736 \pm 0.0059}$ | $\mathbf{0.6888 \pm 0.0146}$ |

Table 5: Repair overlay on natural failures in archived official BFCL outputs. No row uses a test-time teacher; beam4 uses checker-based selection. Rewrite rows use seeds 17/23/31; local repair is a single-run diagnostic. Observed harm (failure on originally correct outputs) is 0.0000 in all rows. Rescue = accuracy conditional on base failure.

| BFCL category | Base | Greedy rewrite | Beam4 rewrite |
|---|---|---|---|
| Multiple calls | 0.7150 | $0.8783 \pm 0.0076$ | $\mathbf{0.9533 \pm 0.0076}$ |
| Parallel calls | 0.4900 | $0.7817 \pm 0.0126$ | $\mathbf{0.9133 \pm 0.0076}$ |
| Parallel multiple calls | 0.4100 | $0.6767 \pm 0.0208$ | $\mathbf{0.8417 \pm 0.0104}$ |
| Simple Java | 0.2700 | $0.5000 \pm 0.0000$ | $\mathbf{0.5533 \pm 0.0058}$ |
| Simple JavaScript | 0.3800 | $0.5867 \pm 0.0231$ | $\mathbf{0.6400 \pm 0.0346}$ |
| Simple Python | 0.7850 | $0.8850 \pm 0.0050$ | $\mathbf{0.9392 \pm 0.0014}$ |

Table 6: Natural-failure transfer by BFCL category. Rewrite rows are 7B students averaged over three seeds with no test-time teacher; beam4 uses checker-based selection. Checker-gated overlay, not raw leaderboard.

## 6 Analysis

**Execution filtering and factorization address different failure modes.** Direct SFT imitates proposals that never fix the trace. Response-level filtering removes those, but an accepted trace can still carry incidental decisions (formatting, call order, irrelevant fields) that have nothing to do with the actual repair. Whole-trace RFT/STaR eliminates the first noise source but not the second. The balanced experiment measures the combined training procedure. The augmented comparison is consistent with a benefit from local targets, but its additional training data prevent attributing the 3.06-point gap to factorization alone.

**Training changes coverage; replay exposes it.** CAR-point, CausalFlow-oracle-step, and RVFD select exactly the same number of successful repairs from a shared candidate pool: the selector is not what produces the train-time gain. The difference is upstream: factor-level supervision makes a successful repair more likely to appear among the student's candidates, even at greedy decoding.

**The best target depends on failure structure.** The factor-trained student performs well on localized errors. Real outputs often break multiple constraints at once, and whole-trace rewriting covers interactions that no single local edit can express. The two are complementary: factors for localized credit assignment, traces for compound errors.

**Proposal coverage is the residual bottleneck.** Under beam16 replay, stop/missing-call success sits at 0.9297, below the other three factors. A termination edit changes both trajectory length and the point at which later actions become available. The tau2 oracle analysis shows the same ceiling at larger scale: no selector over the current candidate set can pass 0.6429. Once coverage runs out, additional replay buys nothing.

## 7 Robustness and Validity Checks

**Training seeds and test examples capture different uncertainty.** The reported standard deviation measures variation across seeds 17, 23, and 31 for a fixed test set. It does not measure uncertainty over new tool schemas or environments. Paired tests compare methods on the same held-out examples; the main table marks seed-17 tests with $p_{17}$, and the appendix reports the per-seed scores.

| Regime | Test-time aid | Success |
|---|---|---|
| Base 7B | none | 0.0893 |
| Clean router | none | 0.5536 |
| Candidate oracle | label oracle | 0.6429 |
| Hybrid executor | replay+fallback | $0.7500 \pm 0.0357$ |

Table 7: Scope boundary on tau2 policy56. No row uses a test-time teacher. The oracle is not deployable; the hybrid uses explicit assistance.

A significant paired difference establishes that the observed success sets differ under this evaluation protocol. It does not identify the cause of the difference when training data or assistance differ. We accordingly interpret significance together with the data and decoding conditions of each comparison, and make no population-wide claim from three training seeds.

**The whole-trace comparison has a training-data confound.** The nearest alternative to RVFD is whole-trace rejection-sampling fine-tuning: generate a repair, execute it, train on traces that succeed. Our augmented comparison fixes the 7B model, holdout, greedy decoding, and seeds, but uses different training files: 1,023 factor rows and 681 whole-trace rows, with 256 and 170 development rows. Per-seed factor-level gains are $+0.0273$, $+0.0293$, and $+0.0352$ for seeds 17, 23, and 31, with paired-bootstrap 95% CIs of $[0.0059, 0.0488]$, $[0.0039, 0.0566]$, and $[0.0098, 0.0605]$; exact McNemar $p = 0.02431$, $0.03570$, $0.007916$. The improvement is consistent across seeds. Statistical significance does not, however, remove the training-data confound; an equal-data rerun is needed to estimate the independent effect of factor granularity.

**Syntactic validity does not imply a useful label.** Of 1024 well-formed DeepSeek proposals, 173 ($16.9\%$) fail replay and are excluded before SFT. On the cross-provider surface, 39 of 512 schema-completed GPT/Claude proposals ($7.6\%$) are rejected. These fractions are not a teacher quality ranking (prompts and procedures differ); they simply quantify the core failure mode: strong models produce plausible edits that execution vetoes.

**Repair gains do not come from indiscriminate rewriting.** Aggregate accuracy can mask a system that fixes failures while quietly breaking correct outputs. We therefore report rescue (accuracy conditional on initial failure) and harm (failure conditional on initial success). On natural BFCL, greedy and beam4 rewriting rescue $0.4661 \pm 0.0081$ and $0.6888 \pm 0.0146$ of failures with observed harm of $0.0000$. The latter follows from the checker gate, which preserves initially correct outputs; it does not measure the safety of unrestricted rewriting. Likewise, zero false adoption under replay is relative to the acceptance checker, not an independent guarantee of real-world correctness.

**Independent splits and natural failures test generalization.** BFCL-CF evaluation is proposal-disjoint; the cross-provider experiment adds source-disjointness. Swapping the main teacher for GPT-5.4 and Claude Sonnet 4.6 leaves the student and acceptance rule unchanged. The natural study runs on archived official BFCL outputs that were generated without knowledge of our corruption operators and scored by the public Gorilla checker. None of these checks alone is conclusive, but together they reduce dependence on any single proposal model, corruption process, or selector.

**Model size does not resolve the supervision bottleneck.** The balanced 7B and 14B students reach similar mean success under each target construction, while both improve with verified factor targets. In this small local-repair setting, increasing capacity alone therefore offers no clear advantage. This is not evidence that larger models cannot help: the proposal set is limited, and neither benchmark coverage nor training-data size scales with model size. The result instead motivates evaluating data construction alongside capacity, with the same held-out tasks and inference budget.

## 8 Related Work

**Tool use and function calling.** Function-calling and interactive-agent benchmarks test whether language models can select APIs, build arguments, and maintain state across actions (Gorilla Team, 2024; tau2-bench Contributors, 2025; ToolSandbox Contributors, 2024; Patil et al., 2023; Yao et al.,

2022b; Schick et al., 2023; Qin et al., 2023; Li et al., 2023; Zhou et al., 2023; Yao et al., 2022a; Shridhar et al., 2021; Wang et al., 2022a; Liu et al., 2023). Most score the final policy or trace. Our question is narrower: what supervision should a student learn from the single local intervention that would have fixed a failed trace?

**Distillation and on-policy training.** Knowledge distillation and preference optimization transfer behavior from larger models or preference signals (Hinton et al., 2015; Kim & Rush, 2016; Hsieh et al., 2023; Agarwal et al., 2024; Rafailov et al., 2023); instruction tuning and synthetic-data pipelines rely on model- or human-generated targets in much the same way (Ouyang et al., 2022; Wang et al., 2022b; Mukherjee et al., 2023). STaR and rejection-sampling fine-tuning retain trajectories that pass an outcome test (Zelikman et al., 2022; Touvron et al., 2023); our whole-trace baseline is exactly this recipe applied to tool repair. On-policy distillation addresses distribution shift via student-generated states, temporal curricula, or rubric-based feedback (OPD Contributors, 2025; Wang et al., 2026; Ye et al., 2025; Fang et al., 2026). RVFD is orthogonal: given a black-box teacher proposal, which local intervention does execution support well enough to serve as supervision?

**Counterfactual repair.** CausalFlow and CAR are the closest prior work: both intervene on agent traces and evaluate counterfactual executions (Bonagiri et al., 2026; Shah, 2026), building on classical counterfactual formulations (Holland, 1986; Rubin, 2005; Pearl, 2009). CausalFlow localizes failure-inducing steps and constructs minimally repaired contrastive pairs; CAR estimates outcome changes under candidate-action interventions. RVFD differs in both granularity and use: interventions act on tool-schema factors, and replay acceptance directly fixes the SFT target that a black-box teacher proposed. The RFT/STaR-style comparison evaluates the full training recipe, with the data-size caveat above; the selector experiment shows that RVFD reaches, but does not surpass, oracle-style CAR and CausalFlow readouts when candidates are held fixed.

## 9 Limitations

RVFD requires resettable execution or a faithful checker. Hidden state, irreversible actions, and proxy judgments weaken its validation guarantee. BFCL-CF uses reference-trace equality, which can reject valid alternatives and cannot certify live API behavior.

Performance is bounded by proposal coverage. The residual stop/missing-call errors and the tau2 candidate oracle both indicate that additional replay cannot compensate for missing successful candidates.

The principal benchmark is constructed from BFCL and deliberately restricts each controlled example to a local repair. The natural-failure experiment reduces dependence on this construction, but remains a checker-gated overlay: it is not an official leaderboard submission and should not be interpreted as one. Similarly, the cross-provider study establishes that the acceptance rule works with more than one proposal source, but its 512 proposals do not capture the full diversity of tools, providers, or interaction styles.

Finally, the free-form tau2 results do not support claims about general autonomous agents. Replay plus structured fallback yields a capable repair executor, but student-only policies and published-method variants remain weak on the stateful policy56 tasks. RVFD is a method for constructing and selecting local repairs, not an end-to-end policy-learning algorithm.

## 10 Conclusion

RVFD turns black-box teacher proposals into local repair supervision by checking each intervention before training. The balanced experiments improve greedy 7B and 14B students without test-time assistance. Augmented factor training also exceeds the whole-trace control, with a data-size confound that requires a controlled follow-up. Natural failures favor broader trace rewriting, and stateful tasks remain difficult for the standalone student. Together, these results support verified local repair as a useful training interface while placing clear limits on claims about factorization and autonomous policy learning. A decisive next test is an equal-data comparison that holds both the accepted examples and the generated candidate budget fixed across repair targets.

## AI USE STATEMENT

The human authors are responsible for the research questions, final methodological choices, and scientific claims. AI tools assisted with implementation, experiment planning and analysis, figure preparation, and manuscript drafting and revision. All AI-generated code was reviewed and verified by the authors. GPT-6 Astra was additionally used to improve the clarity and readability of the manuscript.

## REFERENCES

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. Causalflow: Causal attribution and counterfactual repair for LLM agent failures, 2026. Counterfactual agent-trace repair baseline.

DeepSeek-AI. DeepSeek-V3 technical report, 2024.

Junfeng Fang, Zhepei Hong, Mao Zheng, Mingyang Song, Gengsheng Li, Houcheng Jiang, Dan Zhang, Haiyun Guo, Xiang Wang, and Tat-Seng Chua. Rubric-based on-policy distillation, 2026.

Gorilla Team. Berkeley Function Calling Leaderboard. https://gorilla.cs.berkeley.edu/leaderboard.html, 2024. Benchmark and non-live function-calling scorer. Accessed: 2026-07-08.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327. Association for Computational Linguistics, 2016.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents, 2023.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of GPT-4, 2023.

OPD Contributors. On-policy distillation for language model agents. https://github.com/THUNLP-MT/OPD, 2025. Implementation reference used for local baseline adaptation. Accessed: 2026-07-08.

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. In *Advances in Neural Information Processing Systems*, 2022.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis, 2023.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009.

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, 2023.

Qwen Team. Qwen2.5: A party of foundation models. https://qwenlm.github.io/blog/qwen2.5/, 2024. Model family documentation. Accessed: 2026-07-08.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

Donald B. Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023.

Jaineet Shah. Causal agent replay: Counterfactual attribution for LLM-agent failures, 2026. Published-code counterfactual agent replay baseline.

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

tau2-bench Contributors. tau2-bench: Stateful tool-use evaluation for language agents. https://github.com/sierra-research/tau2-bench, 2025. Software benchmark. Accessed: 2026-07-08.

ToolSandbox Contributors. ToolSandbox: A stateful benchmark for tool-using language agents. https://github.com/apple/ToolSandbox, 2024. Software benchmark. Accessed: 2026-07-08.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. TCOD: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents, 2026.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Cote, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader?, 2022a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2022b.

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*, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2022b.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents, 2023.

# A Reproducibility Index

This appendix collects data, training, and evaluation details for the reported results. The anonymous artifact includes training files, source code, compact evaluation records, and aggregate statistics. Model weights are not included. Some historical seed-17 per-example logs are unavailable; the corresponding per-seed aggregates and paired discordant counts are retained.

| Artifact category | Contents |
|---|---|
| BFCL-CF construction records | Corruption generation, factor labels, holdout splits |
| Teacher proposal tables | Accepted/rejected local repairs with replay outcomes |
| Student training | Full-parameter 7B/14B training code and seed metadata |
| Evaluation logs | Greedy, beam, replay-selection scores |
| Published-baseline adapters | CAR, OPD, TCOD, and GKD reproduction interfaces |
| Stateful stress-test records | tau2 policy56 tasks, budgets, per-seed outcomes |
| Runtime/version manifest | Package versions, public-code commits, model configs |

Table 8: Reproducibility index for the anonymous code and data artifact.

# B BFCL-CF Data Card

BFCL-CF is a local-repair benchmark derived from BFCL (Gorilla Team, 2024). Each example begins with a canonical tool-call trace and corrupts exactly one schema factor. A repair is accepted only when applying the edit and replaying the modified trace recovers the reference outcome.

**Corruption pool.** The diagnostic pool contains 11,773 factor-level corruptions: 3,276 argument-key, 3,276 argument-value, 1,919 function, and 3,302 stop-call instances. Function corruptions substitute another legal in-schema API; key corruptions rename an argument; value corruptions insert a deterministic incorrect value; stop-call corruptions delete a required action or terminate the trace prematurely.

**Main proposal surface.** The balanced DeepSeek-chat proposal set contains 1,024 records. Replay accepts 851 proposals, partitioned into 681 training and 170 development examples. The accepted set comprises 254 argument-key, 209 argument-value, 252 function, and 136 stop-call repairs.

**Holdouts.** The main holdout contains 512 examples (128 per factor) and shares no proposal identifiers with training. The cross-provider holdout is additionally source-disjoint. Evaluation applies the edit and compares the resulting canonical trace with the reference. This is a reference-based check, not a live API execution or a comparison with the teacher's prose.

# C Prompt and Repair Schema

The teacher receives a failed trace, the factor to repair, and the tool schema, then returns a compact JSON object. Prompt wording varies across providers, but the response schema is fixed:

```
{
  "op": "replace|rename_arg|replace_arg_value|insert_call",
  "step": <integer>,
  "factor": "function|arg_key|arg_value|stop_call",
  "old": <JSON value or null>,
  "new": <JSON value or null>
}
```

The accepted student target is the edit object, without a prose rationale. Canonicalization strips formatting variation and checks schema legality; a candidate enters the training set only after replay succeeds.

**Algorithm 2** Replay acceptance (used for both labeling and test-time selection)

**Require:** failed trace $\tau$, candidate edit $e$, checker $R$
reject if $e$ is not parseable JSON
reject if $e$ modifies a factor outside the requested category
reject if $e$ violates the function schema
$\hat{\tau} \leftarrow \mathrm{Apply}(\tau, e)$
reject if $\hat{\tau}$ is identical to $\tau$
**if** $R(\hat{\tau}) = 1$ **then**
    accept $e$
**else**
    reject $e$
**end if**

## D Training and Runtime Details

Students are full-parameter Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct (Qwen Team, 2024). Training uses FSDP full-shard auto-wrap on Qwen2DecoderLayer, bf16 mixed precision, AdamW with learning rate $10^{-5}$, and three epochs. All experiments use seeds 17, 23, and 31 unless noted otherwise.

| Component | Version / provenance |
|---|---|
| Remote GPU runtime | 8×A100-SXM-80GB, CUDA 12.8 |
| PyTorch | 2.8.0+cu128 |
| Transformers | 4.57.6 |
| Datasets | 4.5.0 |
| Accelerate | 1.10.1 |
| TRL / PEFT | 0.24.0 / 0.17.1 |
| tau2-bench checkout | `d8e915f7` |
| CAR checkout | `6bf4fe36` |
| CausalFlow checkout | `97994cf9` |

Table 9: Runtime environment. Commit hashes are abbreviated; full SHAs appear in the code archive.

## E Per-Seed Main BFCL-CF Results

The main paper reports means and standard deviations across three seeds. Table 10 provides the underlying per-seed measurements for the greedy saved-student comparison; aggregates match Table 2 in the main paper.

| Student | Labels | Seed 17 | Seed 23 | Seed 31 |
|---|---|---|---|---|
| 7B | Direct teacher JSON | 0.7734 | 0.7578 | 0.7656 |
| 7B | Response-level accepted | 0.7715 | 0.7754 | 0.7930 |
| 7B | Replay factor label | 0.8125 | 0.8223 | 0.8262 |
| 14B | Direct teacher JSON | 0.7754 | 0.7617 | 0.7617 |
| 14B | Response-level accepted | 0.7793 | 0.7734 | 0.7734 |
| 14B | Replay factor label | 0.8379 | 0.8086 | 0.8125 |

Table 10: Per-seed greedy BFCL-CF saved-student results. No test-time teacher, verifier, or replay selection is applied.

## F Augmented Factor and Whole-Trace Baselines

The comparison fixes the 7B student, seeds, holdout, and greedy decode. An audit of the released training files finds 1,023 factor rows versus 681 whole-trace rows, and 256 versus 170 development rows. These are different training recipes, so the observed difference cannot isolate supervision granularity. The historical artifact calls this block "matched"; the table below preserves its scores while correcting that interpretation.

| Training target | Seed 17 | Seed 23 | Seed 31 | Mean $\pm$ std |
|---|---|---|---|---|
| Whole-trace RFT/STaR | 0.8984 | 0.8887 | 0.8809 | $0.8893 \pm 0.0088$ |
| Replay factor label | 0.9258 | 0.9180 | 0.9160 | $\mathbf{0.9199 \pm 0.0052}$ |

Table 11: Augmented 7B greedy comparison. Neither row uses a test-time teacher or verifier.

Per-seed factor-minus-trace deltas are $+0.0273$, $+0.0293$, and $+0.0352$. Paired bootstrap 95% intervals are $[0.0059, 0.0488]$, $[0.0039, 0.0566]$, and $[0.0098, 0.0605]$; exact McNemar tests give $p = 0.024$, $0.036$, and $0.008$.

## G Verifier Cost–Quality Trade-off

| Decode / selector | Seed 17 | Seed 23 | Seed 31 | Calls |
|---|---|---|---|---|
| Greedy, no verifier | 0.9258 | 0.9180 | 0.9160 | 0.00 |
| Beam-4 + replay | 0.9629 | 0.9590 | 0.9531 | 3.75 |
| Beam-16 + replay | 0.9746 | 0.9785 | 0.9668 | 12.70 |

Table 12: Per-seed augmented 7B cost–quality results. Replay-selected beams incur zero false adoption; "Calls" is the mean replay invocations per example.

Stop/missing-call is the weakest factor at beam-16, reaching $0.9297 \pm 0.0078$. Unlike replacing an argument, adding or removing a call changes both trajectory length and temporal position, making the search space harder to cover with fixed beam width.

## H Published Causal Selector Reproductions

We run CAR from commit `6bf4fe36`; all 40 released smoke tests pass. CAR-CI applies the published confidence-interval rule, whereas CAR-point uses the point estimate. The CausalFlow adapter is based on commit `97994cf9`. It retains the released trace and graph structures, mapping BFCL-CF steps to minimal interventions over a shared candidate pool. This design isolates step attribution but does not constitute an end-to-end reproduction of CausalFlow proposal generation.

| Selector | Beam-4 | Beam-16 |
|---|---|---|
| CAR-CI | $0.2858 \pm 0.0176$ | $0.6829 \pm 0.0637$ |
| CAR-point | $0.9583 \pm 0.0049$ | $0.9733 \pm 0.0060$ |
| CausalFlow-first | $0.9258 \pm 0.0020$ | $0.9238 \pm 0.0000$ |
| CausalFlow-oracle-step | $0.9583 \pm 0.0049$ | $0.9733 \pm 0.0060$ |
| RVFD replay | $0.9583 \pm 0.0049$ | $0.9733 \pm 0.0060$ |

Table 13: Causal selector comparison on the same 7B candidate pools and replay budgets. All selectors produce zero false adoption on this surface. RVFD ties the oracle-step ceiling while requiring no causal graph.

RVFD improves on CausalFlow-first by 2.93–5.47 points across seeds and budgets, with no CausalFlow-only wins. It ties CausalFlow-oracle-step and CAR-point exactly, locating the selection ceiling. The augmented training-recipe comparison appears in Table 11, with its data-size caveat.

## I DPO Preference Baseline

The DPO baseline learns preferences between accepted and rejected candidates on the same BFCL-CF surface. It reaches 0.6050 with greedy decoding and 0.7050 with beam-4 plus replay. The gap to replay-filtered SFT indicates that pairwise preference supervision does not substitute for the exact executable edit on this task: the DPO model learns *which* candidate looks better but not *how* to produce a correct one from scratch.

## J Cross-Provider Proposal Study

The cross-provider study trains exclusively on proposals from GPT-5.4 and Claude Sonnet 4.6, with no DeepSeek rows in training. Deterministic schema completion followed by replay accepts 473 of 512 proposals. A Qwen2.5-7B student trained on these rows achieves the following on a fresh example- and source-disjoint holdout:

| **Decode / selector** | **Seed 17** | **Seed 23** | **Seed 31** |
|---|---|---|---|
| Greedy, no verifier | 0.8555 | 0.8613 | 0.8613 |
| Beam-4 + replay | 0.9258 | 0.9316 | 0.9395 |

Table 14: Cross-provider 7B student results. The student sees only GPT/Claude proposals during training; evaluation uses a disjoint holdout.

These numbers confirm that replay verification decouples label quality from teacher identity: once replay accepts an edit, the provider that generated it is irrelevant to student performance.

## K Natural BFCL Official-Output Failures

This evaluation applies the trained student as a repair overlay on 1,150 AST rows from archived official BFCL v4 non-live outputs of Qwen2.5-0.5B-Instruct-FC. The Gorilla checker determines whether the base output needs repair and evaluates the resulting candidate.

| **Policy** | **AST acc.** | **Rescue** | **Harm** |
|---|---|---|---|
| Base official output | 0.5939 | – | – |
| Factor local repair | 0.6774 | 0.2056 | 0.0000 |
| Whole-trace greedy | $0.7832 \pm 0.0033$ | $0.4661 \pm 0.0081$ | 0.0000 |
| Whole-trace beam-4 | $0.8736 \pm 0.0059$ | $0.6888 \pm 0.0146$ | 0.0000 |

Table 15: Checker-gated repair of natural official-output failures. Rescue is conditional accuracy on base failures; Harm is conditional failure on base successes. Zero harm follows from preserving checker-accepted base outputs.

| Category | Base | Greedy | Beam-4 |
|---|---|---|---|
| Multiple | 0.7150 | $0.8783 \pm 0.0076$ | $0.9533 \pm 0.0076$ |
| Parallel | 0.4900 | $0.7817 \pm 0.0126$ | $0.9133 \pm 0.0076$ |
| Parallel multiple | 0.4100 | $0.6767 \pm 0.0208$ | $0.8417 \pm 0.0104$ |
| Simple Java | 0.2700 | $0.5000 \pm 0.0000$ | $0.5533 \pm 0.0058$ |
| Simple JavaScript | 0.3800 | $0.5867 \pm 0.0231$ | $0.6400 \pm 0.0346$ |
| Simple Python | 0.7850 | $0.8850 \pm 0.0050$ | $0.9392 \pm 0.0014$ |

Table 16: Natural BFCL AST accuracy by category. Greedy and Beam-4 columns use the whole-trace rewrite student.

## L TAU2 FREE-FORM PROTOCOL

The tau2 policy56 evaluation contains 56 held-out airline, retail, and telecom tasks. Fair comparison requires matching the task set, decode budget, teacher access, and verifier budget. We therefore separate student-only, CAR/replay, routed, and structured-fallback regimes.

| Method | Decode / budget | Test teacher | Test verifier | Seeds | Success |
|---|---|---|---|---|---|
| Base Qwen2.5-7B greedy | greedy | no | no | single | 0.0893 |
| 7B full-policy SFT | greedy ctx4096 | no | no | single | 0.1429 |
| 7B full-policy SFT | beam-4 | no | tool-error replay | single | 0.2143 |
| 14B trace-SFT v24 | greedy | no | no | single | 0.3036 |
| Router v8 clean surface | domain routed | no | mixed/limited | single | 0.5536 |
| Router v9 diagnostic | domain routed | no | mixed/limited | single | 0.5714 |
| CAR-CI | beam-4 | no | yes | 17/23/31 | $0.0893 \pm 0.0000$ |
| CAR-point | beam-4 | no | yes | 17/23/31 | $0.0893 \pm 0.0000$ |
| OPD-family single round | greedy | no | no | 17/23/31 | $0.0179 \pm 0.0000$ |
| TCOD f2b | greedy | no | no | 17/23/31 | $0.0179 \pm 0.0000$ |
| TCOD b2f | greedy | no | no | 17/23/31 | $0.0714 \pm 0.0000$ |
| TRL-GKD $\lambda{=}0$ variant | greedy | no | no | 17/23/31 | $0.0893 \pm 0.0000$ |
| Collected-candidate oracle | all collected | no | label oracle | diagnostic | 0.6429 |
| Structured-critical hybrid | beam-2+fallback | no | replay+fallback | 17/23/31 | $0.7500 \pm 0.0357$ |

Table 17: Full tau2 policy56 comparison. The structured hybrid row combines repair execution with deterministic fallback and is not a raw student-only result. OPD/TCOD are protocol adaptations; GKD uses a $\lambda{=}0$ control after the faithful on-policy path exceeded the available FSDP memory budget.

The success oracle inspects the official task label before selecting an output and is therefore not deployable. Its $36/56$ ceiling shows that improving the selector alone cannot lift the current non-API candidate pool above $0.70$.

| Hybrid lookup-0 | Seed 17 | Seed 23 | Seed 31 |
|---|---|---|---|
| Success | 0.7500 | 0.7143 | 0.7857 |
| Tool errors | 22 | 25 | 15 |
| Parse errors | 19 | 21 | 19 |
| Invalid actions | 35 | 34 | 39 |

Table 18: Per-seed tau2 structured-critical hybrid (lookup budget 0). No test-time teacher is used; the regime combines replay verification with deterministic structured fallback.

## M LIMITATIONS

RVFD requires executable replay or a faithful checker. It is a poor fit when effects are irreversible, state cannot be reset, or success can only be judged through ungrounded natural-language descriptions. Beam-4 and beam-16 trade additional execution cost for higher repair accuracy. Free-form tau2 remains weak without structured fallback: the evidence supports local tool-trace repair supervision, not general-purpose autonomous-agent learning.