τ-IF: A MULTI-VARIANT BENCHMARK FOR AGENTIC INSTRUCTION FOLLOWING IN EXECUTABLE ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agents must complete user goals while obeying standing and task-specific instructions. Their actions can alter state and activate different instructions. Existing agent benchmarks rarely pair executable training data with pre-specified instruction variants for testing this joint capability. To this end, we introduce τ-IF, a benchmark built on executable τ-bench environments. τ-IF releases 2,000 verified training programs with corresponding trajectories and provides two evaluation sets: Test evaluates 164 base tasks using one instruction variant each, whereas Test-Variants evaluates the same tasks under all five variants. Each instance combines a user goal, multi-turn interaction, executable tools, mutable environment state, and machine-scored instruction constraints. Across 23 eligible systems on Test, the best observed Task-and-Instruction Success is 40.24%, although Task and Instruction Success Rate scores reach 73.17% and 75.61%, exposing a joint-success shortfall. We then use τ-IF for an optimizer-paired study across three backbones. Full-trajectory SFT exceeds Fixed-Context GRPO and PPO by 5.08 and 4.80 points, while Online variants add $+0.61$ and $+0.16$ points over Fixed. τ-IF connects open training data, broad capability evaluation, and controlled experimentation in one reproducible platform, establishing verified trajectory supervision as a strong baseline while motivating better rewards and credit assignment.

1 INTRODUCTION

LLM agents must obey standing and task-specific instructions throughout tool-mediated interaction, not merely produce a compliant final response. Because actions alter the environment and future context, instruction following is inseparable from the trajectory producing the final state (Yao et al., 2023; 2025; Qi et al., 2025a). Task success can hide policy, parameter, workflow, or communication violations, while constraint pass rates can reward non-completion; explicit constraints can also impede task solving (Qi et al., 2026). Evaluation must therefore combine executable outcomes with explicit trajectory-level instruction verification.

We introduce τ-IF, an executable benchmark and data suite built on τ-bench Retail and Airline (Yao et al., 2025). It releases 2,000 verified Retail programs and trajectories, Dev-200, and two evaluation sets over 164 base tasks: Test freezes one pre-specified instruction variant per task, whereas Test-Variants contains all five. Each instance combines a user goal, evolving interaction, tools, mutable state, and structured constraints with automated verdicts. Its primary metric, *Task-and-Instruction Success* (TIS), requires Task Success—the requested outcome—and *Instruction Success Rate* (ISR), which passes only when every active instruction check does. *Constraint Satisfaction Rate* (CSR) instead averages individual active-check verdicts. Test ranks 23 eligible systems, while Test-Variants evaluates robustness to instruction variation. Across heterogeneous audited protocols, the best TIS point estimate is 40.24%, despite Task and ISR maxima of 73.17% and 75.61%. High component scores therefore do not guarantee joint success.

We compare verified full-trajectory SFT with RL over frozen decision histories or live user–tool trajectories across three backbones and 21 runs. SFT scores 5.08/4.80 TIS points above the implemented fixed-context GRPO and PPO recipes; Online adds only $+0.61/+0.16$ points over Fixed. The released data, metrics, ranking, and study provide a reproducible foundation for agentic instruc-

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.

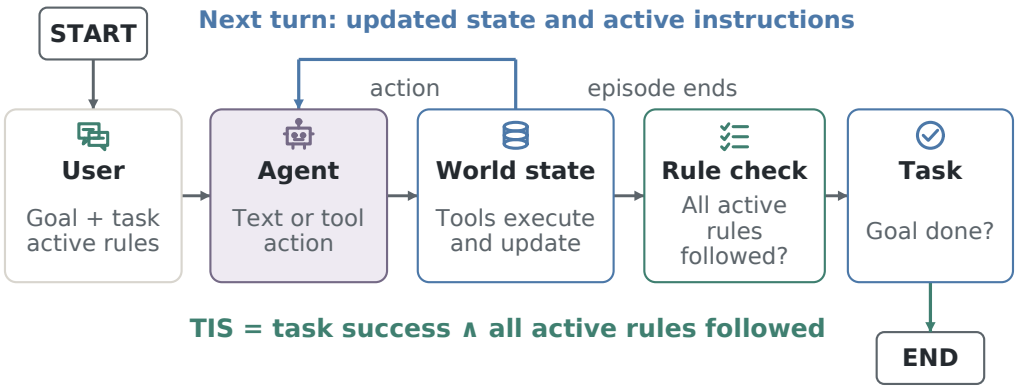

Figure 1: τ-IF runtime and terminal scoring. User context conditions an action; the resulting world state determines the next observation and active rules. Nonterminal trajectories loop. At termination, TIS requires task success and compliance with every active rule.

tion following in executable environments. Because evaluation and training share the same task semantics, capability ranking, variant robustness, and recipe comparisons remain directly connected. Figure 1 summarizes the runtime loop and joint terminal criterion.

2 Related Work

Instruction-following benchmarks. IFEval, FollowBench, InFoBench, and ComplexBench evaluate deterministic, fine-grained, or compositional constraints (Zhou et al., 2023; Jiang et al., 2024; Qin et al., 2024a; Wen et al., 2024). Multi-IF extends across turns and languages, while AgentIF and OctoBench target agentic tasks (He et al., 2024; Qi et al., 2025a; Ding et al., 2026). These suites expose instruction failures but do not jointly combine trajectory-level applicability, mutable executable outcomes, released training trajectories, and frozen task variants within one reproducible protocol for direct comparison.

Executable agent evaluation. ReAct couples reasoning with actions; AgentBench, API-Bank, ToolLLM, and the Berkeley Function-Calling Leaderboard (BFCL) evaluate agent or function-calling ability (Yao et al., 2023; Liu et al., 2024; Li et al., 2023; Qin et al., 2024b; Patil et al., 2025). WebArena, AppWorld, and ToolSandbox provide executable, stateful worlds (Zhou et al., 2024; Trivedi et al., 2024; Lu et al., 2025). τ-bench and τ^2-bench add tool–agent–user interaction (Yao et al., 2025; Barres et al., 2026), while newer benchmarks stress long horizons and execution noise (Shen et al., 2026; Wang et al., 2026). τ-IF additionally scores task–instruction conjunction without claiming unexposed base tasks.

Demonstrations, policy internalization, and online learning. Toolformer learns API use; AgentTuning and InstructGPT study demonstration-based tuning and human-feedback learning; CAP-CPT internalizes policy documents and combines with SFT (Schick et al., 2023; Zeng et al., 2024; Ouyang et al., 2022; Liu et al., 2026b). Policy-induced shift motivates supervision on visited states (Ross et al., 2011), while RL work studies reward optimization, online curricula, and multi-turn agents (Shao et al., 2024; Qi et al., 2025b; Wang et al., 2025; Feng et al., 2025; Qian et al., 2025). We compare these recipes on one executable task pool.

Evaluation reliability and documentation. LLM-judge biases, documentation practice, and contamination risks motivate explicit provenance, split boundaries, and exposure caveats (Zheng et al., 2023; Gebru et al., 2021; Deng et al., 2024). We therefore release frozen prompts, task-level inputs, executable artifacts, and row-level protocol provenance, while treating judge dependence and public base-task exposure as limitations.

3 THE τ-IF BENCHMARK

We represent a constraint-augmented task as

$$\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, P, \mathcal{C}),$$

where $\mathcal{S}$, $\mathcal{O}$, and $\mathcal{A}$ are the state, observation, and action spaces; P gives the distribution over the next state and observation after an action; and $\mathcal{C} = \{c_1, \ldots, c_K\}$ is the task's collection of instruction checks. Each check represents one requirement and specifies when it applies, which agent actions it governs, and how the complete interaction history is examined to determine whether the requirement was followed. States encode the user goal and mutable environment, observations contain instructions and tool results, and actions are text, tool calls, or termination. Because actions affect state and check activation, identical final database states can have different instruction outcomes. Parameter rules can depend on whether particular arguments are supplied, whereas confirmation, ordering, and budget rules can span turns.

For task i, let τ_i be the retained trajectory and $t_i \in \{0, 1\}$ its executable task-success verdict; let $m_{ij}, v_{ij} \in \{0, 1\}$ be check j's activation and pass verdict evaluated from τ_i. We define

$$\mathrm{ISR}_i = \prod_{j:m_{ij}=1} v_{ij}, \qquad \mathrm{TIS}_i = t_i\,\mathrm{ISR}_i, \qquad \mathrm{CSR} = \frac{\sum_{i,j} m_{ij} v_{ij}}{\sum_{i,j} m_{ij}}.$$

ISR is an all-or-nothing task-level measure: it is one exactly when every active check passes. TIS additionally requires $t_i = 1$. CSR instead micro-averages active verdicts over task–check pairs, but its denominator is model-dependent because actions affect m_{ij}, so CSR is diagnostic. A tool-targeted check activates when its target tool is called; checks without a target remain active, and inactive checks do not enter the product. The empty product is one, but every reported Test and Test-Variants item has always-active checks.

Task verdicts t_i come from executable state and action checks, whereas v_{ij} comes from a frozen gpt-oss-120b trajectory judge rather than human annotation; the latter are benchmark measurements with known judge limitations (Zheng et al., 2023; OpenAI, 2025). A cancellation can yield $t_i = 1, \mathrm{ISR}_i = 0$ after violating confirmation or ordering; compliance without task completion can yield the reverse. Both have $\mathrm{TIS}_i = 0$. Appendix A formalizes applicability, responsibility, empty-set cases, serving configuration, and recipe rewards.

4 BENCHMARK CONSTRUCTION AND RELEASE

4.1 AGENTIC IF TASK SYNTHESIS

We synthesize structured Retail task programs specifying a user request, required and forbidden state changes, information requirements, multi-turn user behavior, a mutable environment, legal tool paths, and terminal conditions. Natural-language requests are rendered only after the executable structure is fixed. Constraints span tool parameters, multi-parameter relations, tool flow, budgets, response handling, global and pointwise scope, and conditional activation. Each program passes static schema and conflict checks, oracle execution, and reference-rollout verification. The accepted set contains 1,718 unmodified Kimi K3 teacher trajectories and 282 bounded repairs produced by GPT-5.6 Sol from frozen traces that already had task success (Kimi Team, 2026; OpenAI, 2026a). Repairs preserve the successful mutation and pass oracle replay, two checks by a separate frozen Hy3 judge (Tencent, 2026a), and independent review; Appendix C records their provenance.

Each Test-Variants task–variant instance contains 72–138 instruction checks (mean 106.1) across the five families, including argument relations, tool order, budgets, and cross-call state. Table 1 counts unique constraint identifiers, not their total occurrences across the 820 Test-Variants cells. ISR is produced by the same frozen trajectory judge defined above, whereas executable state and action checks determine Task independently for every reported task.

4.2 TRAJECTORY DATA CURATION

Routing follows rejected $>$ conditional $>$ clean. A decision is *clean* only if it triggers neither a hard-rejection nor a conditional rule. *Conditional* decisions are plausible but ambiguous, insufficiently verified, repair-affected, or downstream of such an event. *Rejected* records are fixed pre-user

Table 1: τ-IF splits and constraints. Panel (a) summarizes units and uses (Test-Variants: 164 tasks $\times$ 5 variants = 820 cells); panel (b) counts unique constraint identifiers by family within each split.

(a) Data splits

Split	Domain	Unit	# Units	Variants/unit	# Instances	Primary use
Train	Retail	program	2,000	1	2,000	Training
Dev	Retail	program	200	1	200	Checkpoint selection
Test	Retail+Airline	base task	164	1	164	Benchmark ranking
Test-Variants	Retail+Airline	base task	164	5	820	Recipe/robustness

(b) Constraint taxonomy (unique identifiers)

Family	Train	Dev	Test-Variants	Constraint target
Single parameter	1,479	276	565	One tool argument
Multi-parameter	539	96	240	Relation among arguments
Tool flow	1,182	77	410	Tool-call sequence
Tool budget	1,223	88	492	Call-count limit
Multi-turn state	975	191	359	Cross-turn state
Total	5,398	728	2,066	–

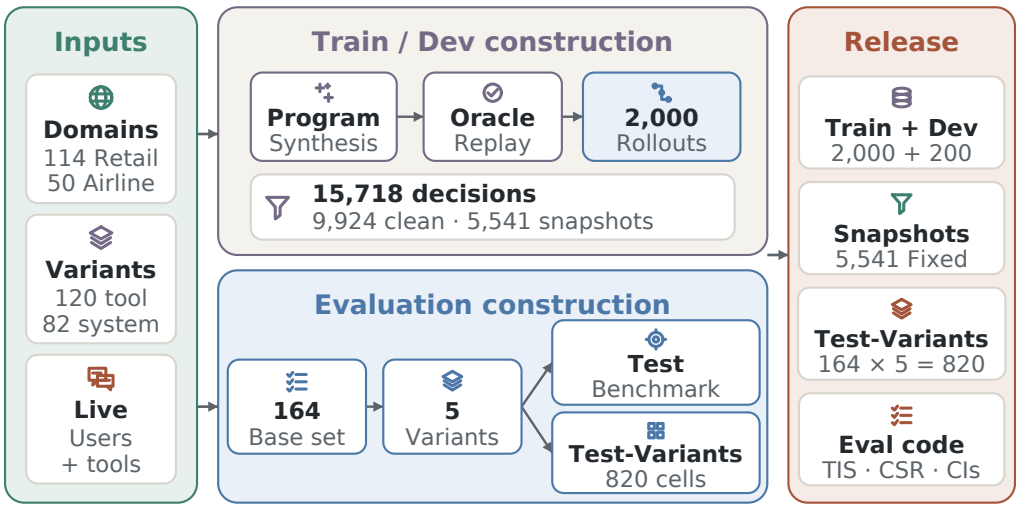

Figure 2: τ-IF construction and release. Verified Retail programs yield 2,000 Train trajectories, Dev-200—the sole checkpoint-selection split—and 5,541 snapshots. Frozen τ-bench overlays yield Test for 23-system benchmark ranking and Test-Variants for recipe comparison and robustness. Test-Variants spans 114 Retail and 50 out-of-domain Airline tasks with 120 tool and 82 system-prompt variants used in evaluation.

greetings or decisions that fail deterministic admissibility checks. No LLM may promote a record to clean. This conservative priority keeps ambiguous or repair-affected decisions out of Fixed-Context training, whose snapshot pool admits only clean tool calls. Complete trajectories retain both text actions and tool calls with their surrounding observations for SFT.

The 15,718 assistant decisions comprise 9,924 clean, 3,163 conditional, and 2,631 rejected decisions. The clean set contains 5,541 tool calls and 4,383 text actions. The release provides two complementary products: 2,000 complete trajectories retaining text, tool calls, observations, and transitions, and 5,541 clean single-tool snapshots—one per clean tool-call decision—from 1,982 programs. Figure 2 places these two training-data products alongside the released Dev, Test, and Test-Variants evaluation resources in one overview.

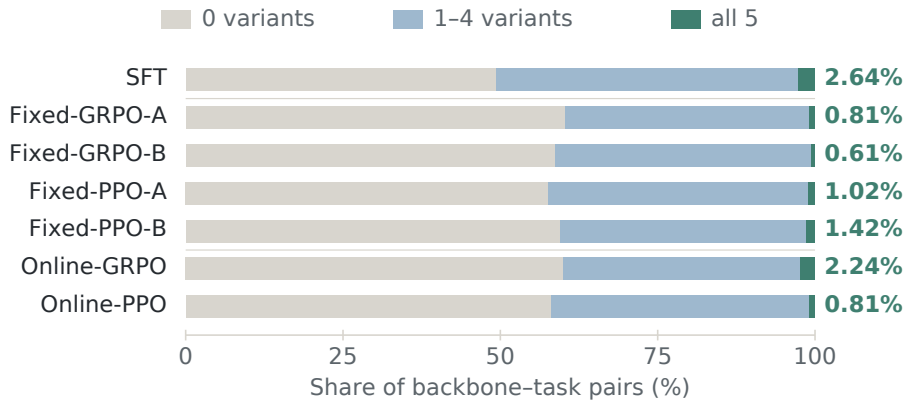

Figure 3: Multi-variant robustness on Test-Variants. Each bar partitions three Dev-selected backbones $\times$ 164 tasks ($n = 492$) by how many frozen variants satisfy TIS: zero, one to four, or all five. Right labels give all-five rates; Appendix K reports full 0–5 counts.

4.3 Multi-variant evaluation design

Train/Dev use newly synthesized Retail programs; Airline is out of distribution by domain. The 164 evaluation tasks comprise 114 Retail and 50 Airline tasks. Test freezes one predefined variant per task for external benchmark ranking; Test-Variants applies all five to the same base tasks. Self-trained checkpoints are selected only on Dev-200 by TIS, ISR, Task, CSR, then earlier training progress. Test's variants are assigned approximately evenly within domain by a fixed mapping released with the data. Each is the corresponding frozen Test-Variants overlay, so constructing Test adds neither model calls nor test-based selection. The evaluation catalog has 120 tool and 82 system-prompt variants. Exact instruction texts duplicated across split-level catalogs are retained once, with priority Test-Variants, Train, then Dev when assigning catalog membership.

Within each five-variant cluster, the goal, environment, and terminal criterion stay fixed while tool-specific or system-prompt instructions change. This isolates robustness to instruction variation from failure on the underlying task.

Evaluation applies bounded retries only to unusable outputs; semantic errors are never resampled, and all 87 conditions end with zero model-output or runtime errors. Appendix D documents recovery paths and protocol exceptions. Uncertainty uses 10,000 fixed-seed base-task bootstrap draws, keeping all five Test-Variants overlays for each resampled task together; Test retains its corresponding single overlay in every draw.

We release all four splits, verified trajectories, 5,541 Fixed-Context snapshots, cluster/variant mappings, schemas, manifests, and evaluation code.

5 Benchmark Evaluation

Test benchmarks executable Task and strict instruction success on a common 164-task slice and ranks the 23 eligible systems by TIS point estimate; this benchmark ranking is descriptive, not a causal comparison of training methods. Table 2 reports both components and their conjunction for every eligible row. Because TIS is a per-task conjunction, aggregate Task and ISR rates do not determine it: success on one cannot compensate for failure on the other, and similar marginal rates can conceal different joint-success rates. The ranking therefore rewards joint success within the same interaction rather than strong but disjoint marginal outcomes across different tasks. The component columns show whether each shortfall comes from goal completion or instruction compliance. The highest Task (73.17%) and ISR (75.61%) belong to different systems. Although Fable 5.1 is not highest on either component, it attains the highest TIS point estimate (40.24%). Service models occupy the strongest observed range, but none dominates both components. Atria Dawn Preview is the foundation agentic model of Guo et al. (2026). Qwen3.5-9B is the strongest Open

Table 2: τ-IF Test point estimates and TIS ranking (%) for 23 systems on the same 164 tasks. Within each group and metric, bold and underlined entries mark first and second. Protocols differ across rows; Appendix I gives exact model identities, sources, recovery provenance, and exceptions. Point-estimate order need not be statistically significant.

Model	Task	ISR	TIS
Service models			
Fable 5.1	64.02	56.71	**40.24**
GPT-6 Astra	57.93	<u>68.29</u>	<u>39.02</u>
GPT-5.6 Sol	64.02	54.88	34.76
Gemini 3.1 Pro	64.63	55.49	32.93
Grok 4.6	42.07	**75.61**	28.66
Qwen3.8-Max	64.63	50.00	29.88
DeepSeek-V4-Pro	67.68	54.88	35.37
Seed2.1 Pro	**73.17**	37.80	25.61
Kimi K3	68.29	53.05	34.76
Hy3	64.02	45.73	26.83
Hy4 preview	71.34	43.90	31.71
Atria Dawn Preview	<u>71.95</u>	37.20	28.66
Open-weight models			
Llama3.1-8B	10.98	10.37	1.83
ToolACE-2.5-8B	17.07	27.44	8.54
Llama-xLAM-2-8B	37.20	29.88	12.20
Ministral-3-8B Reasoning	18.90	27.44	6.10
gpt-oss-20b	12.20	**48.78**	6.71
Qwen3.5-9B	<u>62.80</u>	39.63	**26.83**
Qwen3.5-27B	**67.07**	39.63	<u>26.22</u>
Gemma 4 12B IT	50.00	<u>40.24</u>	21.34
Ours			
Llama3.1-8B SFT q3	18.90	27.44	5.49
Qwen3.5-9B SFT q4	<u>55.49</u>	**51.83**	<u>28.05</u>
Qwen3.5-27B SFT q1	**65.24**	<u>45.12</u>	**31.10**

row (26.83% TIS) (Qwen Team, 2026a), and Gemma 4 12B IT the strongest non-backbone open baseline (21.34%) (Gemma Team, 2026); ToolACE and xLAM are tool-specialized references (Liu et al., 2025; Prabhakar et al., 2025).

For Qwen3.5-9B/27B, SFT changes ISR by $+12.20/+5.49$ percentage points (pp) and Task by $-7.31/-1.83\,pp$; TIS is 28.05/31.10% versus Initial's 26.83/26.22%, without a detectable paired gain. The cross-family non-thinking Llama stress test rises from 1.83% to 5.49% TIS after SFT but remains low overall. These are tool-capable instruction-tuned references before τ-IF adaptation, not pretrained base models. Component profiles differ: Seed2.1 Pro has the highest Task but 37.80% ISR, whereas Grok 4.6 has the highest ISR but 42.07% Task; both remain below 30% TIS. The TIS point estimates define the benchmark ordering, but overlapping marginal intervals do not support a statistically significant total ordering. Matched SFT–Initial paired intervals appear in Appendix I.

For Ours, q1–q4 denote quarter-progress checkpoints selected on Dev. External systems were not run on Test-Variants; they are therefore not directly compared with the training-recipe results in Table 3. Appendix I reports row-level recovery and protocol exceptions for every evaluated row.

6 Training-Recipe Study

6.1 Training methods

We use τ-IF to compare full-trajectory SFT, Fixed-Context RL, and Online Environment RL. We ask whether SFT scores above Fixed and whether Online scores above its optimizer-matched Fixed counterpart in TIS. We also test whether Online-minus-Fixed varies with the gold solution's tool-action count or number of distinct write tools, and whether aggregates conceal Task/ISR, domain,

variant, or checkpoint differences. These task features are outcome-independent. The primary contrast is Online minus Fixed within backbone and optimizer; cross-backbone summaries are descriptive. TIS is primary, while Task and ISR distinguish goal completion from instruction compliance.

Full-trajectory SFT. We apply standard SFT to complete agent trajectories (Ouyang et al., 2022; Zeng et al., 2024). Training uses all 2,000 verified trajectories after removing the fixed opening from each. The loss is computed only on assistant text and tool-call tokens, not on system instructions, user messages, or tool responses. Qwen uses thinking-compatible traces; Llama uses its native non-thinking protocol. Retaining the preceding interaction context lets supervision cover both language and state-conditioned action choices.

Fixed-Context RL. Fixed-Context RL uses the 5,541 clean tool-call snapshots selected by deterministic filtering. At each frozen context, the current policy produces a group of candidate actions that are scored without execution. Candidate actions neither mutate stored state nor produce a next observation, so snapshots remain independent. Variant A combines deterministic tool/schema gates with a constraint-only judge; B adds a reasonableness judge for multiple valid actions. GRPO uses group-relative advantages and KL-divergence regularization. PPO treats each decision as terminal with $R = r$ and $A = r - V(x)$ and uses clipped policy and value objectives (Shao et al., 2024; Schulman et al., 2017; Qian et al., 2025).

Online Environment RL. Online Environment RL samples from the same training-program pool as the Fixed-Context methods but does not freeze the interaction history. The policy interacts turn by turn with a simulated user and executable tools; tool calls update the database, so subsequent contexts and applicable instruction constraints depend on earlier policy actions. At the end of each trajectory, the terminal reward equally weights Task Success and the satisfaction rate over active constraints; only checks activated on the realized trajectory enter that satisfaction term. Online GRPO computes group-relative advantages across groups of complete trajectories, whereas Online PPO uses generalized advantage estimation (GAE) to assign multi-turn returns to individual assistant decisions (Schulman et al., 2016; Qi et al., 2025b; Wang et al., 2025).

Controls. To align the training-recipe comparisons, methods share the initial model, training-program pool, instruction constraints, and Dev checkpoint-selection rule; Fixed and Online methods are paired within GRPO and PPO. SFT learns from text and tool actions in complete trajectories, Fixed RL uses clean one-step tool-call snapshots, and Online RL receives feedback from complete environment trajectories. Appendix B summarizes state source, action type, feedback, and parameter updates for each method.

6.2 Experimental design

We train Qwen3.5-9B/27B and Meta-Llama-3.1-8B-Instruct (Qwen Team, 2026a; Llama Team, 2024); Qwen uses thinking and Llama is a fixed non-thinking cross-family stress test. Each backbone has seven runs: SFT, Fixed-GRPO A/B, Fixed-PPO A/B, Online-GRPO, and Online-PPO. Four quarter-progress checkpoints yield 84 trained conditions; three Initial conditions yield 87. The A/B Fixed runs vary the reward definition and are paired variants rather than independent training seeds. All 87 conditions complete evaluation on Dev-200 and Test-Variants. Evaluation uses bounded retries; in earlier evaluations, some unusable responses triggered a restart of the entire task. Appendix D reports the frequency and impact of these restarts.

Selection is nested within each run and never compares candidates on either Test or Test-Variants. One Dev-selected checkpoint per run is evaluated on all 820 Test-Variants cells. Instruction variants measure how performance changes across instruction configurations for the same task, not repeated training. The task bootstrap resamples the 164 fixed base tasks with replacement to quantify sensitivity to benchmark task composition, not variation across training seeds. All five variants of a sampled base task move together, so rewrites of one goal are not treated as independent observations; the base task remains the unit of uncertainty.

SFT uses full-parameter fine-tuning, whereas RL uses LoRA (Hu et al., 2022); each training run uses eight GPUs. The release includes complete hyperparameter configurations and checkpoint slots, including optimizer and batch/group settings, sequence limits, and training steps.

Table 3: Dev-selected Test-Variants TIS (%) by run using final retained records. Bold and underlined entries mark first and second per backbone.

Method	Qwen3.5-9B	Qwen3.5-27B	Llama3.1-8B
Initial	24.3	22.3	1.0
SFT	**27.4**	**30.5**	**5.5**
Fixed-GRPO-A	24.5	22.4	0.9
Fixed-GRPO-B	23.9	23.1	1.6
Fixed-PPO-A	23.1	<u>24.3</u>	<u>1.7</u>
Fixed-PPO-B	<u>25.1</u>	23.1	0.9
Online-GRPO	24.8	24.2	1.1
Online-PPO	24.2	23.7	<u>1.7</u>

7 Training-Recipe Results

7.1 Full-trajectory SFT is highest among implemented recipes

Table 3 reports Dev-selected TIS for all recipes. SFT has the highest TIS on each backbone, while RL stays near Initial. Cells are runs; clustered intervals reflect benchmark task composition. Appendices F and D give component metrics and recovery results.

All 12 backbone-by-reward-variant SFT-minus-Fixed contrasts are positive. The descriptive cross-backbone/A–B means are $+5.08\,pp$ for GRPO, with a 95% confidence interval (CI) of $[+3.25, +7.01]$, and $+4.80\,pp$ for PPO ($[+3.05, +6.63]$); nine of 12 stratified intervals exclude zero (Table 17). The difference is concentrated in ISR: SFT-minus-Fixed changes Task/ISR by $-1.02/+11.54\,pp$ for GRPO and $-2.62/+13.31\,pp$ for PPO. Appendix J reports paired success transitions for every condition.

7.2 Online interaction yields no detectable average gain

The matched Online recipes show no detectable average TIS improvement: Online-minus-Fixed is $+0.61\,pp$ for GRPO (95% task-cluster CI $[-0.85, +2.15]$) and $+0.16\,pp$ for PPO ($[-1.12, +1.38]$); all 12 stratified intervals cross zero (Table 17). The optimizer-paired contrasts appear in Figure 4. A Qwen-only sensitivity check gives $+0.98\,pp$ for GRPO and $+0.03\,pp$ for PPO; both intervals span zero within this subset.

7.3 Online interaction enriches training signals

All 18 RL runs show optimization signals. For Qwen GRPO, groups with zero reward standard deviation fall from 97.99% in Fixed to 1.95% in Online (mean standard deviation 0.156). This richer feedback coincides with the highest mean Task score but no consistent TIS gain: Online reward averages Task and active-check CSR, whereas TIS requires their conjunction. Reward dispersion alone therefore does not align optimization with the joint objective. Appendices E and L report run-level dynamics and the lack of a consistent relationship between Online-minus-Fixed and the gold solution's tool-action or distinct-write-tool count.

7.4 Performance varies across instruction variants and domains

Across SFT, Fixed, and Online, SFT leads mean TIS, ISR, and all five diagnostic active-verdict family rates (Table 12). Yet all-five TIS is only 2.64%, 0.97%, and 1.52%, respectively; many tasks pass none (Figure 3; Appendix K). This stricter statistic complements mean TIS by measuring cross-variant consistency for each base task.

Dev-200 is synthesized Retail; Test-Variants adds 50 out-of-domain Airline tasks to 114 Retail tasks (Table 11). Dev selection matches the retrospective Test-Variants-best checkpoint in 7 of 21 runs; main results retain Dev-only selection (Appendix H).

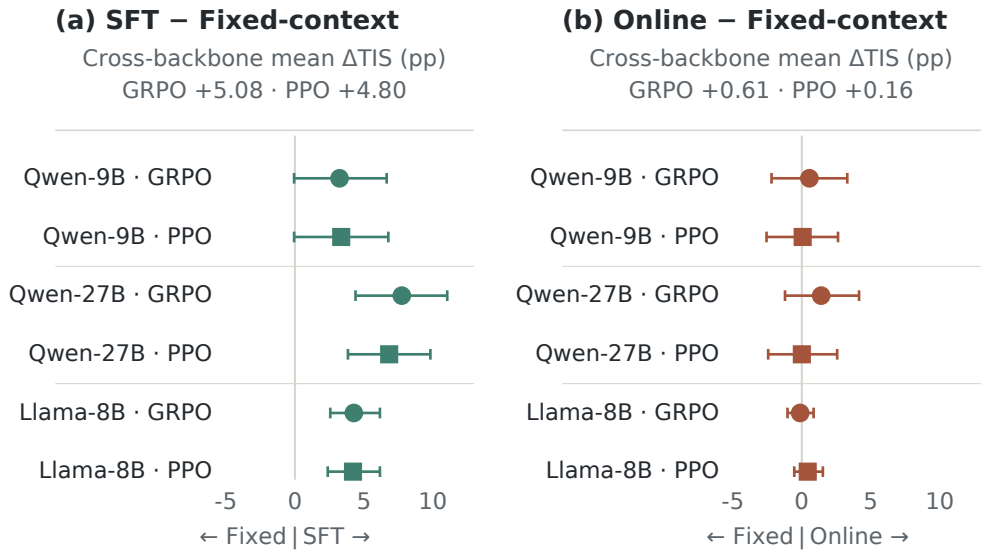

Figure 4: Optimizer-paired TIS differences in percentage points (pp) with 95% task-cluster bootstrap intervals. Positive values favor SFT in (a) and Online in (b); circles/squares denote GRPO/PPO.

8 Discussion and Limitations

Task and ISR are distinct: neither guarantees TIS, and their leaders differ on Test. Low all-five success exposes variant brittleness. Online yields richer rewards, yet SFT has higher TIS. Evaluation should report joint/component metrics, variants, and activation rules; Appendix A reassesses CSR.

The Online results distinguish reward richness from objective alignment: more varied feedback does not by itself guarantee joint success. A promising direction is to retain that interaction signal while assigning credit more directly to completion-plus-compliance events, then test whether gains persist across instruction variants and domains. This would target the observed gap without discarding the benefits of environment interaction. Because Test and Test-Variants share base tasks, this alignment can be checked at the level of individual goals rather than only aggregate means. Such analysis can distinguish broadly improved policies from gains confined to one instruction realization, task family, or evaluation domain, providing a concrete target for future reward comparisons.

Limitations include single runs, Retail-only Train/Dev, 164 clusters, no human or second-judge calibration, public τ-bench exposure risk (Deng et al., 2024), heterogeneous recovery, incomplete compute/training measurements, and recipe differences beyond interaction. TIS is not a safety rate; protocol differences and overlapping intervals limit claims beyond point-estimate ranking. Human calibration and repeated runs would strengthen the estimates.

9 Conclusion

We introduce τ-IF, an executable benchmark and data suite with verified Train/Dev programs and one- and five-variant evaluations over 164 tasks. Task, ISR, CSR, and TIS separate goal completion from instruction compliance. Test ranks 23 systems and exposes a Task–instruction gap; Test-Variants reveals low cross-variant consistency.

Across three backbones and two optimizers, verified-trajectory SFT is strongest; Online yields richer rewards without a detectable gain over matched Fixed. τ-IF unifies benchmark ranking, multi-variant robustness, and reproducible recipe comparison in executable environments. Together, these results establish verified trajectory supervision as a strong reference point and motivate training objectives aligned with joint task-and-instruction success.

AI USE STATEMENT

Generative AI was used as a supporting tool, not as the originator of the research question, hypotheses, experimental design, analysis plan, or scientific claims. The authors conceived the study, specified all experimental protocols, reviewed the data and model outputs, determined the analyses, and manually revised and approved every part of the manuscript. Within the research pipeline, AI models rendered pre-specified synthetic tasks and constraints, generated teacher trajectories and bounded trajectory repairs (Section 4.1), and served as user simulators and judges. AI tools also assisted with code implementation and debugging, literature retrieval, reference formatting, vector-figure implementation, and preliminary drafting, translation, and editing of the manuscript. No generative image model was used. All AI-assisted artifacts were checked through deterministic validation, executable replay, frozen tests, targeted human auditing, and manual review. The authors take full responsibility for the final data, code, text, figures, and claims.

ETHICS STATEMENT

All users, orders, and database entities in training are synthetic. Tool agents can nevertheless cause unauthorized state changes, expose private information, or exploit ambiguous confirmation; deployment requires authentication, confirmation, least privilege, logging, and human escalation. Project-generated data are released under Creative Commons Attribution 4.0 (CC BY 4.0) and project-authored code under the MIT License; upstream τ-bench/Tau2 MIT notices are preserved.

REPRODUCIBILITY STATEMENT

The released τ-IF artifact includes Train, Dev, Test, Test-Variants, verified trajectories, Fixed-Context snapshots, schemas, cluster/variant mappings, release manifests, and evaluation code. It also includes configurations for all 21 training runs, the 84-checkpoint registry, recovery records, Test bootstrap inputs and rebuild scripts, the activation-sensitivity ledger, a path-redacted compute ledger, fixed analysis seeds, task-level tables, and figure scripts. Four large JSON Lines (JSONL) members use lossless XZ transport compression with a standard-library, hash-verifying materializer; Test remains directly readable and its extraction makes no model calls.

REFERENCES

Alibaba Cloud. Qwen3.8-Max, 2026. URL https://www.alibabacloud.com/help/en/model-studio/qwen3-8-max. Official model documentation.

Anthropic. Claude Fable 5.1 & Claude Mythos 5.1 System Card, September 2026. URL https://www-cdn.anthropic.com/0339e6a7c5c7b87f5c07798616dc32c215d14235/Claude%20Fable%205.1%20%26%20Claude%20Mythos%205.1%20System%20Card.pdf. Published September 1, 2026.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/oral/71171.

ByteDance Seed. Seed2.1 Model Card: Agentic intelligence for productivity, June 2026. URL https://seed.bytedance.com/en/seed2_1. Official model card for Seed2.1 Pro and Seed2.1 Turbo.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. arXiv preprint arXiv:2606.19348, 2026. URL https://arxiv.org/abs/2606.19348.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 8706–8719, 2024. doi: 10.18653/v1/2024.naacl-long.482. URL https://aclanthology.org/2024.naacl-long.482/.

Deming Ding, Shichun Liu, Enhui Yang, Jiahang Lin, Ziying Chen, Shihan Dou, Honglin Guo, Weiyu Cheng, Pengyu Zhao, Chengjun Xiao, Qunhong Zeng, Qi Zhang, Xuanjing Huang, Qidi Xu, and Tao Gui. OctoBench: Benchmarking scaffold-aware instruction following in repository-grounded agentic coding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5958–5978, 2026. doi: 10.18653/v1/2026.acl-long.269. URL https://aclanthology.org/2026.acl-long.269/.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/420c9f777c0b4f78d515e53cf74d58b2-Abstract-Conference.html.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64 (12):86–92, 2021. doi: 10.1145/3458723.

Gemma Team. Gemma 4 technical report. arXiv preprint arXiv:2607.02770, 2026. URL https://arxiv.org/abs/2607.02770.

Google. Gemini 3.1 Pro Preview, February 2026. URL https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview. Official model documentation for gemini-3.1-pro-preview.

Google DeepMind. Gemma 4 12B IT model card, 2026. URL https://huggingface.co/google/gemma-4-12B-it. Hugging Face model card.

Honglin Guo, Tao Gui, et al. Atria Dawn: The dawn of agentic superintelligence. arXiv preprint arXiv:2609.15818, 2026. URL https://arxiv.org/abs/2609.15818.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. Multi-IF: Benchmarking LLMs on multi-turn and multilingual instructions following. arXiv preprint arXiv:2410.15553, 2024. URL https://arxiv.org/abs/2410.15553.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4667–4688, 2024. doi: 10.18653/v1/2024.acl-long.257. URL https://aclanthology.org/2024.acl-long.257/.

Kimi Team. Kimi K3: Open frontier intelligence. arXiv preprint arXiv:2607.24653, 2026. URL https://arxiv.org/abs/2607.24653.

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*, pp. 3102–3116, 2023. doi: 10.18653/v1/2023.emnlp-main.187. URL https://aclanthology.org/2023.emnlp-main.187/.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, et al. Ministral 3. arXiv preprint arXiv:2601.08584, 2026a. URL https://arxiv.org/abs/2601.08584.

Jiateng Liu, Zhenhailong Wang, Xiaojiang Huang, Yingjie Li, Xiang Li, Chenlei Guo, Xing Fan, Ruhi Sarikaya, and Heng Ji. Analyzing and internalizing complex policy documents for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16814–16851, 2026b. doi: 10.18653/v1/2026.acl-long.767. URL https://aclanthology.org/2026.acl-long.767/.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8EB8k6DdCU.

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. In *International Conference on Learning Representations*, pp. 52989–53046, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ.

Llama Team. The Llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024. URL https://arxiv.org/abs/2407.21783.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Meta. Llama 3.1 8B Instruct model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct. Hugging Face model card.

Mistral AI. Ministral-3-8B-Reasoning-2512 model card, 2025. URL https://huggingface.co/mistralai/Ministral-3-8B-Reasoning-2512. Hugging Face model card.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card. arXiv preprint arXiv:2508.10925, 2025. URL https://arxiv.org/abs/2508.10925.

OpenAI. GPT-5.6 System Card, July 2026a. URL https://deploymentsafety.openai.com/gpt-5-6. Published July 9, 2026; covers GPT-5.6 Sol, Terra, and Luna.

OpenAI. GPT-6 Astra System Card, September 2026b. URL https://deploymentsafety.openai.com/gpt-6-astra/gpt-6-astra.pdf. Published September 3, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract.html.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, et al. APIGen-MT: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. arXiv preprint arXiv:2504.03601, 2025. URL https://arxiv.org/abs/2504.03601.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Amy Xin, Youfeng Liu, Bin Xu, Lei Hou, and Juanzi Li. AgentIF: Benchmarking large language models instruction following ability in agentic scenarios. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-1892. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/51bb3a8a33610a25aae074bfc51b1b1f-Abstract-Datasets_and_Benchmarks_Track.html.

Yunjia Qi, Hao Peng, Xintong Shi, Amy Xin, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. On the paradoxical interference between instruction-following and task solving. arXiv preprint arXiv:2601.22047, 2026. URL https://arxiv.org/abs/2601.22047.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c66e1fcc9691aae706250638f36f681b-Abstract-Conference.html.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/97c5b2707228e7e3fb67e4ecc2e0e607-Abstract-Conference.html.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13025–13048, 2024a. doi: 10.18653/v1/2024.findings-acl.772. URL https://aclanthology.org/2024.findings-acl.772/.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, pp. 9695–9717, 2024b. URL https://openreview.net/forum?id=dHng2O0Jjr.

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

Qwen Team. Qwen3.5-27B model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.5-27B. Hugging Face model card.

Qwen Team. Qwen3.5-9B model card, 2026c. URL https://huggingface.co/Qwen/Qwen3.5-9B. Hugging Face model card.

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

Salesforce AI Research. Llama-xLAM-2-8b-fc-r model card, 2025. URL https://huggingface.co/Salesforce/Llama-xLAM-2-8b-fc-r. Hugging Face model card.

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. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

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

Yuanzhe Shen, Zisu Huang, Zhengyuan Wang, Muzhao Tian, Zhengkang Guo, Chenyang Zhang, Shuaiyu Zhou, Zengjie Hu, Dailin Li, Jingwen Xu, Kaimin Wang, Wenhao Liu, Tianlong Li, Fengpeng Yue, Feng Hong, Cao Liu, and Ke Zeng. TRIP-Bench: A benchmark for long-horizon interactive agents in real-world scenarios. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62985.

SpaceXAI. Model card: Grok 4.6, August 2026. URL https://media.x.ai/v1/website/card-4p6-4cd2dc57.pdf.

Team-ACE. ToolACE-2.5-Llama-3.1-8B model card, 2025. URL https://huggingface.co/Team-ACE/ToolACE-2.5-Llama-3.1-8B. Hugging Face model card.

Tencent. Hy3 model card, 2026a. URL https://huggingface.co/tencent/Hy3. Hugging Face model card.

Tencent. Tencent releases and open-sources Tencent Hy4 Preview, August 2026b. URL https://www.tencent.com/tencent-releases-and-open-sources-tencent-hy4-preview/.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Ruipeng Wang, Yuxin Chen, Yukai Wang, Chang Wu, Junfeng Fang, Xiaodong Cai, Qi Gu, Hui Su, An Zhang, Xiang Wang, Xunliang Cai, and Tat-Seng Chua. AgentNoiseBench: Benchmarking robustness of tool-using LLM agents under noisy condition. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ORVJCJczoG.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. arXiv preprint arXiv:2504.20073, 2025. URL https://arxiv.org/abs/2504.20073.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmarking complex instruction-following with multiple constraints composition. In *Advances in Neural Information Processing Systems: Datasets and Benchmarks Track*, volume 37, pp. 137610–137645, 2024. doi: 10.52202/079017-4371. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f8c24b08b96a08ec7a7a975feea7777e-Abstract-Datasets_and_Benchmarks_Track.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Z.ai. GLM-5.3: Frontier coding with emergent cyber capabilities, 2026. URL https://z.ai/blog/glm-5.3. Official model announcement.

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. doi: 10.18653/v1/2024.findings-acl.181. URL https://aclanthology.org/2024.findings-acl.181/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. arXiv preprint arXiv:2311.07911, 2023. URL https://arxiv.org/abs/2311.07911.

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. In *International Conference on Learning Representations*, pp. 15585–15606, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

A Formalization and Objectives

For task i and instruction check j, let $t_i \in \{0,1\}$ denote executable task success, $m_{ij} \in \{0,1\}$ activation, and $v_{ij} \in \{0,1\}$ the check verdict. We report

$$\mathrm{CSR} = \frac{\sum_i \sum_j m_{ij} v_{ij}}{\sum_i \sum_j m_{ij}}, \qquad \mathrm{ISR}_i = \prod_{j:m_{ij}=1} v_{ij}, \qquad \mathrm{TIS}_i = t_i \,\mathrm{ISR}_i.$$

Tool-targeted checks are active when the target tool is called; checks without a target tool are always active. Uncalled tool-targeted checks pass vacuously. The empty product is therefore one: a task with no active check is retained, its ISR is one, and its TIS reduces to TaskSuccess. CSR is micro-averaged only over active checks. Because activation depends partly on model actions, CSR and active-check family slices are diagnostic rather than fixed-denominator causal outcomes. In Test-Variants, every item has 3–11 always-active system-level checks, so all task variants have a nonempty active set. Mean active verdicts per task are 37.41 for SFT, 37.68 for Fixed, and 37.09 for Online (37.19 for Initial). Release metadata's activation and local-decidability fields are descriptive annotations, not additional programmatic scorers.

Frozen scorer configuration. Instruction verdicts use a frozen serving stack exposed under the `gpt-oss-120b` identifier. OpenAI documents the public `gpt-oss-120b` checkpoint as an open-weight 116.8B-parameter mixture-of-experts model with 5.1B active parameters per token (OpenAI, 2025); we cite the card for architecture. Constraint scoring uses prompt `taubench-constraints-reason-first-v3` and at most 20 checks per batch. The evaluator schema requests temperature 0 and top-p 1, but the frozen serving transport used temperature 0.7 and top-p 0.95. The parser requires every supplied constraint identifier exactly once with a nonempty evidence field and Boolean verdict; invalid responses receive corrective feedback and bounded schema retries. Natural-language assertions use the same frozen stack. The user simulator uses that stack separately at requested temperature 1 with a fixed role guard. The `gpt-oss-20b` row in Table 2 is a distinct, smaller evaluated checkpoint, not the judge.

Frozen constraint-judge prompt. For reproducibility, the complete frozen system prompt is reproduced below. The prompt requests JSON output; line breaks are presentational only.

```
You are a strict evaluator of tool-call PARAMETER constraint
following. Judge every supplied constraint against the complete
ordered agent trajectory. Use tool results only where grounding or
```

```
derivation requires them. Do not judge whether the business task was
solved; the official TauBench evaluator scores that separately.

Rules:
1. Evaluate cross-call ordering, count, distinctness, consistency,
   grounding, conditionals, and final-answer constraints across the
   complete trajectory.
2. Follow each requirement and rubric exactly.
3. If the target tool never appears, apply the existing benchmark
   convention: the tool-specific constraint passes vacuously unless
   its own rubric explicitly says otherwise.
4. For every constraint, generate a non-empty evidence-based reason
   FIRST and only then the boolean pass decision. Verdict object key
   order must be exactly: id, reason, pass.
5. Return every supplied constraint id exactly once. Output JSON only.
```

The user message is a compact JSON serialization with the following fields and required output structure:

```
{
  "trajectory": [...],
  "tool_results": [...],
  "final_answer": "...",
  "constraints": [...],
  "required_output": {
    "verdicts": [{
      "id": "<constraint_id>",
      "reason": "<analyze evidence first>",
      "pass": true
    }]
  }
}
```

A conservative offline applicability audit recomputes activation from stored tool arguments without changing frozen verdicts. Across 37,080 task rows (87×200 Dev plus 24×820 selected Test-Variants rows) it removes 308,636 legacy-active checks, almost all vacuous passes, while retaining 200,425 ambiguous checks as active rather than guessing. Eleven ISR rows and eight TIS rows change, but none of the 21 Dev checkpoint winners changes. Among the 24 selected Test-Variants conditions, only 8 of 19,680 task rows change TIS, condition-level TIS moves by at most $0.12\,pp$, and CSR decreases by 0.88–$3.00\,pp$. Because the audit is deliberately conservative and leaves ambiguous triggers unresolved, we retain the frozen primary protocol and release the full sensitivity ledger.

Training rewards are not evaluation TIS. For retained context x, let $g(x,a)$ be the deterministic parse/tool/schema gate, $c(x,a)$ the binary frozen constraint-judge verdict, and $q(x,a)$ the binary conjunction of the reasonableness dimensions; $a^\star$ is the canonical verified target call. Fixed reward A is $r_A = g(x,a)c(x,a)$. Fixed reward B is $r_B = g(x,a)c(x,a)[\mathbb{1}(a = a^\star) + \mathbb{1}(a \neq a^\star)q(x,a)]$. Online terminal reward is $r_{\text{on}} = \frac{1}{2}\text{TaskSuccess} + \frac{1}{2}\text{CSR}_{\text{active}}$, with the active-constraint term set to one when no constraint activates. Thus reward variation and strict TIS are deliberately distinct quantities.

Fixed-Context PPO terminates after one assistant action, so $R = r(x,a)$ and $A = R - V(x)$. Online PPO uses trajectory returns and GAE. GRPO normalizes returns within action groups for Fixed and trajectory groups for Online.

B Training-Recipe Comparison

For a verified target $a^\star$, SFT minimizes $-\log \pi_\theta(a^\star \mid x)$. Fixed RL maximizes expected verifier reward $r_{\text{fixed}}(x,a)$ without executing a, whereas Online RL optimizes trajectory return under policy-induced transitions. The resulting recipes differ in more than environment access:

Table 4: Training-recipe design differences. Fixed and Online are paired by optimizer, but the recipes also differ in state and action support, feedback horizon, environment execution, and update scope.

	Full-trajectory SFT	Fixed-Context RL	Online RL
State source	verified trajectory	frozen decision history	policy-induced trajectory
Action support	text + tool actions	clean tool calls	text + tool actions
Feedback	token-level target	one-step verifier reward	trajectory reward
Environment step	no	no	yes
Parameter update	full	LoRA	LoRA

C Data Construction and Provenance

The 2,000/200 Train/Dev programs are separated at program level. Each program contains a newly instantiated task world, executable user objective, multi-turn interaction policy, tools, and instruction checks. Held-out Test and Test-Variants task bodies, states, constraint identifiers, and check definitions are not used to generate training instances. Public policies and tool schemas remain a benchmark-exposure limitation.

Teacher trajectories and bounded repairs have separate provenance. Unmodified rollouts are Kimi K3 traces that already pass the frozen trajectory judge. GPT-5.6 Sol generated the bounded repairs. They cannot replace the successful task mutation; every accepted repair is oracle-replayed, rechecked twice by a separately frozen Hy3 judge, and independently reviewed. For auditability, we retain the failed verdict from before repair and record the patch operations and rationale. The deterministic decision router records hard rejection and conditional reasons. Table 5 gives an abridged view of one actual released Fixed-Context training record.

Table 5: Abridged view of released snapshot `train_retail_v04_program_000560_aee88d77e8::turn::001`. Displayed identifiers are synthetic; the email, full prompt, tool schemas, and complete check arrays are shortened for readability but remain intact in the release.

Field	Released-record content
Data product	Clean Fixed-Context snapshot from `data/fixed/snapshots.jsonl`.
Observed context	System policy and 16 executable tool schemas, followed by: "Please cancel the order—it's #W3169009. My account email is *[synthetic email]*."
Target decision	Assistant calls `find_user_id_by_email` with the supplied email; no environment mutation occurs at this snapshot.
Instruction metadata	44 active checks; two email-lookup checks are locally decidable at this turn, with their responsibility windows fixed to the current decision.
Routing metadata	The route is `clean`, reason tags are empty, and the source trajectory passed; the record is admitted to the 5,541-snapshot training pool.

D Evaluation and Recovery Protocol

The intended same-step protocol retries only an unusable model step and keeps history, tools, and prompt fixed. The first usable output is accepted without inspecting Task or constraint scores; incorrect actions and parameters are not retried. A complete offline audit covers all 174 result files, 88,740 task histories, and 280,767 retained attempts. It finds 15 same-step recoveries (all depth one; no exhaustion) and 4,729 legacy whole-task reopens, with seven tasks appearing in both classes. Reopens improve TIS for 757 tasks and leave 3,972 unchanged; none decrease it. Latest-row summaries undercount recovery-affected tasks by 4,722.

Whole-task reopens are new episodes and do not preserve the original history. Same-step retries share one client loop, but no prompt/history hash was retained. Score blindness is supported by the audited code path rather than an independent runtime attestation. We therefore report the two mechanisms separately.

Table 6: Attempt-level recovery provenance. Earliest TIS uses each task's earliest scoreable record; Final TIS includes outcomes after all historical reruns.

Scope	Histories	Empty	Same-step	Reopen	Improved	Earliest TIS	Final TIS
All Dev+Test-Variants	88,740	4,737	15	4,729	757	11.60	15.87
Selected Test-Variants	19,680	1,415	0	1,415	244	13.82	16.87

For the earliest-scoreable-record sensitivity analysis, unusable outputs are skipped, but the first scoreable record is retained regardless of its Task or instruction score. We then rerun Dev selection and Test-Variants analysis offline. Six of 21 checkpoint winners change; no selection group requires the unavailable CSR tie-breaker. SFT-minus-Fixed remains $+7.22\,pp$ for GRPO (95% CI $[+5.18, +9.35]$) and $+6.46\,pp$ for PPO ($[+4.61, +8.33]$). Online-minus-Fixed is $-0.22\,pp$ ($[-1.99, +1.59]$) and $+1.18\,pp$ ($[-0.22, +2.52]$), respectively. Thus the qualitative recipe-level conclusions are unchanged.

Infrastructure-only interrupted episodes may be restored only on predeclared task keys with replacement provenance. Fable 5.1 is an explicit exception: only an initial call plus two retries were available, and 10 persistent empty/refusal cases are scored zero.

E Implementation and Compute Details

SFT uses full-parameter updates; all RL conditions use all-linear LoRA adapters. Methods otherwise use matched model-specific templates and fixed tool serialization. Qwen uses a thinking-capable protocol; Llama is non-thinking. Fixed GRPO uses group size eight. Fixed PPO uses a single-step value head and old log probabilities. Online variants execute the same program distribution in isolated environments. Training configurations, software requirements, and checkpoint slots are available in the release.

Table 7: Training hyperparameters; slash-separated values follow the Qwen3.5-9B/Qwen3.5-27B/Llama3.1-8B order. G is samples per group and I is policy iterations per sample set. The length cap is total sequence length for SFT and generated completion length for RL.

Recipe	Epochs	Learning rate	G	I	KL β	Length cap
SFT	4	4e-6/2e-6/4e-6	–	–	–	32k/32k/16k
Fixed GRPO A/B	1	1e-5/8e-6/1e-6	8	1	.01/.01/.01	1024/1024/512
Fixed PPO A/B	1	1e-6/2e-7/2e-7	1	4/2/2	.01/.10/.10	1024/1024/512
Online GRPO	1	2e-6/2e-6/2e-7	4	1	.10/.10/.20	12288
Online PPO	1	1e-6/1e-6/2e-7	1	4/4/2	.01/.01/.10	12288

All runs use seed 42, bfloat16, one example per device, and eight GPUs. SFT uses full-parameter tuning with accumulation four; RL uses all-linear LoRA rank/alpha 32/64 and accumulation 16. Qwen RL samples at temperature 0.9 and top-p 1.0. Llama uses 0.9/1.0 for Fixed and 0.2/0.9 for Online. These differences are part of the implemented recipes. Training and evaluation both contain judge-mediated components; no judge-independence ablation is claimed.

Compute-ledger audit. All 21 training runs record optimizer steps and eight-device allocations; complete logical-run trainer time is available for 19. Rollout counts are reconstructable for all 18 RL runs. Exact training tokens are available for 0/21 runs, however, and training-time verifier and environment-call counts are available for 0/12 Fixed and 0/6 Online runs. GPU type and shared-service allocation intervals are also missing. We therefore leave cross-method GPU-hours null and make no token, wall-time, or efficiency comparison.

A latest-scored-record scan covers all 174 cells in the 87-condition evaluation matrix: 88,740 episodes contain 1,136,082 policy requests, 15.48 billion policy input tokens, 301.45 million policy output tokens, 756,289 environment tool calls, and 201,046 verifier calls. Input-token totals sum per-request usage and therefore recount growing conversation context; they are evaluation workload,

not unique-corpus or training-token counts. Evaluation wall time is unavailable. Early Dev winners remain a selection-stability observation, not a realizable predeclared stopping policy.

Training-log audit. The authoritative checkpoint registry resolves 21/21 training runs and 84/84 checkpoint slots. SFT loss and token accuracy improve for all three backbones. No RL run is flagged as having no update: each records nontrivial gradient and KL or clip signals. Online-GRPO has a mean group reward standard deviation of 0.156 and only 1.95% zero-variance groups. By contrast, the four Qwen Fixed-Context-GRPO runs average 97.99% zero-variance groups, indicating near-collapse of the group-relative advantage signal. All nine PPO runs used `log_entropy=false`, so logged zero entropy values are unavailable measurements rather than evidence of deterministic policies; seven contain extreme explained-variance values. Isolated KL spikes occur in Qwen3.5-9B Online-PPO and Llama Online-GRPO. The audit therefore verifies execution and parameter updates, but not uniformly healthy or well-calibrated RL optimization. Legal-action rates and a positive-control probe were not logged.

F Complete Training-Recipe Results

The following tables retain Task, ISR, CSR, reward variant, and selected checkpoint slot for every training-recipe row. TIS intervals resample 164 base-task clusters and keep all five instruction variants together.

Table 8: Qwen3.5-9B Dev-selected Test-Variants results (%) under final bounded recovery. TIS intervals resample 164 base-task clusters, keeping each task's five variants together.

Method	Reward	Slot	TIS [95% CI]	Task	ISR	CSR
Initial	–	–	24.27 [20.37,28.41]	66.34	35.85	96.19
SFT	–	q4	27.44 [22.93,32.07]	57.44	47.93	97.37
Fixed-GRPO	A	q1	24.51 [20.37,28.78]	63.90	41.34	96.28
Fixed-GRPO	B	q1	23.90 [20.12,27.80]	65.24	38.78	96.45
Fixed-PPO	A	q4	23.05 [19.27,26.95]	69.27	37.32	96.17
Fixed-PPO	B	q3	25.12 [21.22,29.15]	67.80	36.10	96.14
Online-GRPO	–	q1	24.76 [20.49,29.27]	68.54	38.41	96.30
Online-PPO	–	q2	24.15 [20.24,28.29]	69.76	36.46	96.13

Table 9: Qwen3.5-27B Dev-selected Test-Variants results (%) under final bounded recovery. TIS intervals resample 164 base-task clusters, keeping each task's five variants together.

Method	Reward	Slot	TIS [95% CI]	Task	ISR	CSR
Initial	–	–	22.32 [18.17,26.71]	65.85	36.83	96.42
SFT	–	q1	30.49 [26.10,35.00]	61.46	48.78	97.46
Fixed-GRPO	A	q1	22.44 [18.78,26.22]	65.98	34.88	96.01
Fixed-GRPO	B	q1	23.05 [19.15,27.07]	66.59	35.12	96.10
Fixed-PPO	A	q2	24.27 [20.49,28.17]	68.29	38.05	96.45
Fixed-PPO	B	q2	23.05 [19.15,27.07]	68.78	37.32	96.34
Online-GRPO	–	q1	24.15 [20.00,28.29]	65.85	39.63	96.49
Online-PPO	–	q4	23.66 [19.88,27.56]	65.24	39.15	96.47

G Recipe Diagnostics by Domain and Constraint Family

Table 11 averages the Dev-selected recipe rows across backbones and applicable reward variants. These descriptive slices do not make Airline a separately selected test set.

Active-verdict family rates in Table 12 are macro-averaged over selected model conditions. Their denominators depend on model actions, so they diagnose observed trajectories rather than define fixed-denominator causal outcomes.

Table 10: Llama3.1-8B Dev-selected Test-Variants results (%) under final bounded recovery. TIS intervals resample 164 base-task clusters, keeping each task's five variants together.

Method	Reward	Slot	TIS [95% CI]	Task	ISR	CSR
Initial	–	–	0.98 [0.37,1.83]	9.76	6.83	83.87
SFT	–	q3	5.49 [3.78,7.32]	19.02	24.02	93.92
Fixed-GRPO	A	q1	0.85 [0.24,1.59]	9.15	10.98	87.80
Fixed-GRPO	B	q4	1.59 [0.73,2.68]	11.10	11.10	89.83
Fixed-PPO	A	q2	1.71 [0.73,2.80]	8.78	6.46	83.79
Fixed-PPO	B	q2	0.85 [0.24,1.46]	8.66	6.34	83.66
Online-GRPO	–	q4	1.10 [0.37,1.95]	9.39	6.22	83.50
Online-PPO	–	q3	1.71 [0.73,2.93]	10.37	8.66	84.05

Table 11: Dev-selected recipe metrics by domain (%), averaged across backbones and applicable reward variants. "All" is the pooled Test-Variants average.

Recipe	Domain	Task	ISR	TIS
SFT	All	45.98	40.24	21.14
SFT	Retail	43.33	42.75	20.47
SFT	Airline	52.00	34.53	22.67
Fixed	All	47.79	27.82	16.20
Fixed	Retail	44.78	30.70	16.75
Fixed	Airline	54.67	21.23	14.93
Online	All	48.19	28.09	16.59
Online	Retail	44.33	30.76	16.58
Online	Airline	57.00	22.00	16.60

H Checkpoint Selection Diagnostics

Winner slots q1/q2/q3/q4 occur 8/5/3/5 times, 18 of 21 Dev trajectories are mixed, and 7 of 21 Dev-selected checkpoints are also highest on Test-Variants. The mean Test-Variants difference between the retrospectively highest-scoring checkpoint and the Dev-selected checkpoint is 0.56 percentage points. The globally Dev-selected representative is Llama SFT q3 (30.50% Dev TIS), but its Test and Test-Variants TIS are both 5.49%. We retain this Dev-selected model rather than substitute a Qwen checkpoint favored by Test-Variants.

I Complete Test Benchmark Ranking

Model sources. Table 2 shortens model names. The service models are Claude Fable 5.1 (Anthropic, 2026), GPT-6 Astra and GPT-5.6 Sol (OpenAI, 2026b;a), Gemini 3.1 Pro Preview (Google, 2026), Grok 4.6 (SpaceXAI, 2026), Qwen3.8-Max (Alibaba Cloud, 2026), DeepSeek-V4-Pro (DeepSeek-AI, 2026), Seed2.1 Pro (ByteDance Seed, 2026), Kimi K3 (Kimi Team, 2026), Hy3 and Hy4 Preview (Tencent, 2026a;b), and Atria Dawn Preview (Guo et al., 2026).

The open rows correspond to `meta-llama/Meta-Llama-3.1-8B-Instruct` (Llama Team, 2024; Meta, 2024), `Team-ACE/ToolACE-2.5-Llama-3.1-8B` (Liu et al., 2025; Team-ACE, 2025), `Salesforce/Llama-xLAM-2-8b-fc-r` (Prabhakar et al., 2025; Salesforce AI Research, 2025), `mistralai/Ministral-3-8B-Reasoning-2512` (Liu et al., 2026a; Mistral AI, 2025), `openai/gpt-oss-20b` and `openai/gpt-oss-120b` (OpenAI, 2025), `Qwen/Qwen3.5-9B` and `Qwen/Qwen3.5-27B` (Qwen Team, 2026a;c;b), and `google/gemma-4-12B-it` (Gemma Team, 2026; Google DeepMind, 2026). Our SFT and RL rows inherit the listed Qwen and Llama backbones.

The eligibility ledger distinguishes Final87 mixed recovery (F87), formal same-step Recovery@3 (R3), score-blind whole-task empty-output merge (HY), zero-trigger numerical equivalence (ZT), audited infrastructure recovery (IR), and explicit protocol exceptions (EX). GLM-5.3 (Z.ai, 2026) is

Table 12: Active-verdict pass rates by constraint family (%), macro-averaged over selected model conditions. Denominators depend on activated checks and therefore on model actions.

Family	Initial	SFT	Fixed	Online
Single parameter	94.72	99.77	96.43	95.05
Multi-parameter	97.58	99.90	98.23	97.50
Tool flow	92.92	94.63	92.66	92.82
Tool budget	79.28	84.49	78.06	79.09
Multi-turn state	92.15	96.72	93.39	91.97

omitted because its task-keyed recovery provenance is not sufficiently frozen. GPT-5.6 Sol is an EX row: its 164-task run finishes without final errors, but one authentication-failed task was replaced without fully frozen score-blind merge provenance. Fable is also an EX row: it uses imported trajectories, local rejudging, one fewer available retry, and 10 persistent empty/refusal cases scored zero. Seed2.1 Pro is EX because its source configuration permitted unlimited usable-output retries, although the audit observed only three isolated same-step reasoning-only recoveries and no task exceeded the Recovery@3 budget. Kimi K3 uses the maximum reasoning setting; "max" is not part of the model name. Hy3 and Hy4 preview use high reasoning effort.

All included scores are reported once in Table 2; the mapping below moves protocol detail out of the main capability table. The machine-readable eligibility ledger preserves row-level evidence and omitted models.

Table 13: Test evaluation and recovery provenance. Initial denotes the three tool-capable starting checkpoints before τ-IF adaptation.

Code	Models	Evaluation provenance
F87	All three Initial rows; all three SFT rows	Final87 bounded recovery with same-step and legacy whole-task paths audited separately.
R3	Kimi K3	Formal bounded same-step empty-output Recovery@3.
HY	gpt-oss-20b	Score-blind merge over predeclared whole-task empty-output replacements.
ZT	Qwen3.8-Max; Gemini 3.1 Pro; Ministral-3-8B Reasoning; Hy3; Hy4 preview; Atria Dawn Preview	Historical Strict@1 runs with zero unusable-output triggers.
IR	GPT-6 Astra; Grok 4.6; DeepSeek-V4-Pro; Gemma 4 12B IT; ToolACE-2.5-8B; Llama-xLAM-2-8B	Audited score-blind recovery of predeclared infrastructure failures.
EX	Fable 5.1; GPT-5.6 Sol; Seed2.1 Pro	Explicit row-specific protocol exceptions described above and in the released ledger.

Marginal Task, ISR, and TIS intervals for all 23 rows are released in the machine-readable Table 2 analysis file. Marginal intervals should not be read as pairwise significance tests. For the three frozen SFT-versus-Initial comparisons, paired resampling of the same 164 tasks gives:

Table 14: Paired Test SFT-minus-Initial TIS differences (percentage points). Intervals resample the 164 shared base tasks.

Backbone	Difference	95% CI
Qwen3.5-9B	$+1.22$	$[-5.49, 7.93]$
Qwen3.5-27B	$+4.88$	$[-2.44, 12.20]$
Llama3.1-8B	$+3.66$	$[0.00, 7.32]$

J SFT-to-Fixed Task-Variant Transitions

Table 15 reports paired transition counts. They demonstrate that Fixed both loses SFT successes and creates some new successes; they do not by themselves identify the mechanism.

Table 15: Paired SFT-versus-Fixed TIS transition counts over the 820 Test-Variants cells per backbone and Fixed condition.

Backbone	Fixed condition	Both TIS	SFT only	Fixed only	Neither
Qwen3.5-9B	GRPO-A	120	105	81	514
Qwen3.5-9B	GRPO-B	124	101	72	523
Qwen3.5-9B	PPO-A	118	107	71	524
Qwen3.5-9B	PPO-B	123	102	83	512
Qwen3.5-27B	GRPO-A	119	131	65	505
Qwen3.5-27B	GRPO-B	124	126	65	505
Qwen3.5-27B	PPO-A	135	115	64	506
Qwen3.5-27B	PPO-B	117	133	72	498
Llama	GRPO-A	2	43	5	770
Llama	GRPO-B	3	42	10	765
Llama	PPO-A	2	43	12	763
Llama	PPO-B	0	45	7	768

K Multi-Variant Distribution

Table 16: Number of backbone–task pairs passing 0–5 instruction variants. Each method contains three backbones and 164 base tasks ($n = 492$).

Method	0	1	2	3	4	5
SFT	243	107	69	30	30	13
Fixed-GRPO-A	297	82	57	32	20	4
Fixed-GRPO-B	289	91	47	50	12	3
Fixed-PPO-A	283	93	60	40	11	5
Fixed-PPO-B	293	76	72	29	15	7
Online-GRPO	295	87	43	42	14	11
Online-PPO	286	89	59	37	17	4

L Statistical Details

All paired intervals resample base-task clusters and carry all five instruction variants and all compared conditions together. The four scientific contrasts were stated before final analysis, but the exact averaging of A/B and backbones was not independently preregistered. We therefore label the four aggregates descriptive. Figure 4 averages A/B within clusters; the full reward-variant contrasts appear in Table 17.

Table 17: Stratified TIS differences (percentage points). Intervals resample 164 base-task clusters; S–F is SFT minus Fixed and O–F is Online minus Fixed.

Backbone	Optimizer	Reward	S–F	95% CI	O–F	95% CI
Qwen3.5-9B	GRPO	A	+2.93	[−0.73,6.71]	+0.24	[−2.93,3.41]
Qwen3.5-9B	GRPO	B	+3.54	[0.00,7.20]	+0.85	[−2.20,3.90]
Qwen3.5-9B	PPO	A	+4.39	[0.85,8.17]	+1.10	[−1.83,3.90]
Qwen3.5-9B	PPO	B	+2.32	[−1.22,5.98]	−0.98	[−3.90,1.95]
Qwen3.5-27B	GRPO	A	+8.05	[4.39,11.83]	+1.71	[−1.22,4.51]
Qwen3.5-27B	GRPO	B	+7.44	[4.02,11.10]	+1.10	[−2.07,4.39]
Qwen3.5-27B	PPO	A	+6.22	[3.29,9.27]	−0.61	[−3.54,2.20]
Qwen3.5-27B	PPO	B	+7.44	[3.66,11.22]	+0.61	[−2.44,3.54]
Llama	GRPO	A	+4.63	[2.80,6.59]	+0.24	[−0.73,1.22]
Llama	GRPO	B	+3.90	[2.07,5.85]	−0.49	[−1.71,0.73]
Llama	PPO	A	+3.78	[1.83,5.85]	0.00	[−1.10,1.22]
Llama	PPO	B	+4.63	[2.80,6.59]	+0.85	[−0.24,2.07]

The two prespecified task features were frozen without reference to model outcomes. Gold tool-action count is grouped as short (≤ 2), medium (3–5), or long (≥ 6); distinct frozen write-tool count is grouped as low (0), medium (1), or high (≥ 2). The ordinary least squares (OLS) regression

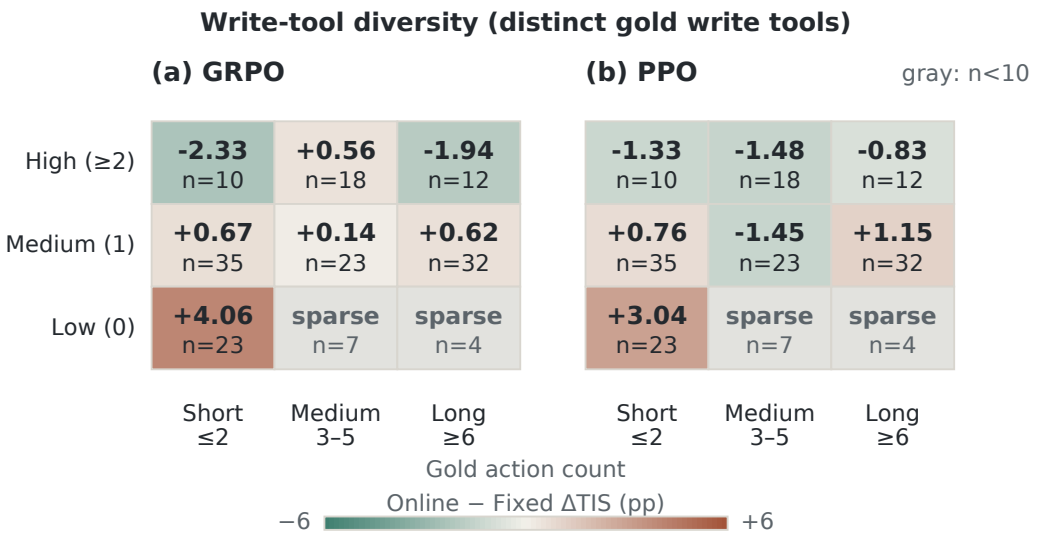

Figure 5: Online-minus-Fixed TIS (percentage points) by two task features defined without reference to model outcomes: gold tool-action count and distinct gold write-tool count. Cells pool backbones and Fixed reward variants and report task-cluster counts; gray cells have $n < 10$ and suppress the effect estimate. Fixed bins and a symmetric scale are retained. A null gradient for these features does not establish that true state dependence is irrelevant.

in Table 18 regresses paired Online-minus-Fixed task deltas on the two standardized counts, their interaction, and optimizer, reward-variant, backbone, and domain indicators. Intervals use the same 10,000-draw base-task cluster bootstrap. Baselines are GRPO, reward A, Qwen3.5-9B, and Retail.

Table 18: OLS coefficients for Online-minus-Fixed TIS (percentage points). Intervals use a 10,000-draw base-task-cluster bootstrap; reference levels are GRPO, reward A, Qwen3.5-9B, and Retail.

Term	Coef.	CI low	CI high
Intercept	+0.16	−2.10	+2.47
Gold actions (standardized)	−0.42	−1.27	+0.41
Distinct write tools (standardized)	−0.71	−1.69	+0.23
Actions × write tools	+0.47	−0.80	+1.54
PPO	−0.45	−2.46	+1.54
Reward B	−0.12	−1.04	+0.83
Qwen3.5-27B	+0.40	−1.98	+2.80
Llama3.1-8B	−0.15	−2.07	+1.74
Airline	+1.00	−1.02	+2.93

Mixed-effects analysis was not run and is not implied by this fixed-effects regression.

M Qualitative Failure Categories

Observed categories include missing explicit confirmation before writes, unverified replacement items, role-reversed simulated users, reasoning-only empty outputs, invalid tool serialization, non-unique order exploration, and over-clarification. These categories motivate deterministic routing and Recovery@3.