SPOT: STATEFUL POST-TRAINING SEARCH WITH LLM RESEARCHERS AND TRAINING DIAGNOSTICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training requires coupled decisions about data, model updates, and checkpoint initialization; because each experiment changes the evidence and checkpoints available next, these decisions form a sequential joint search. LLM researchers can close this loop, but existing systems either expose a narrow action space or permit pipeline changes that make experiments difficult to compare. We present SPOT (**S**tateful **PO**st-**T**raining Search), which converts hypotheses into structured joint interventions and executes them through a shared interface. SPOT records both the proposed recipe and the operation actually consumed by training, then returns its telemetry, evaluation outcomes, and resulting checkpoint to guide subsequent proposals. In tool-use post-training of Qwen3-8B, six search campaigns evaluate 141 candidate checkpoints; every LLM researcher improves over the shared base, and the best search-time composite score increases from 0.4092 to 0.5152 on a 0–1 scale. Interface ablations further show that allowing only one recipe component to change at a time alters both the chosen training method and whether training restarts from the base model or continues from a previous checkpoint. Beyond endpoint scores, the records support three diagnostic questions: do rewards distinguish correct from incorrect outputs; does increasing reward correspond to better tool use; and can the allowed data and training choices address persistent failures? These questions organize retrospective evidence about when to continue training, change sampling, redesign the objective, or expand the available interventions.

1 INTRODUCTION

Post-training adapts a pretrained language model to the tasks and behavior required by a target application (Lambert et al., 2024; Guo et al., 2025). Building it requires coupled choices about supervision construction and allocation (Liu et al., 2024c; Prabhakar et al., 2025; Liu et al., 2024b), update procedure and strength (Rafailov et al., 2023; Zelikman et al., 2022; Gulcehre et al., 2023; Ahmadian et al., 2024; Qian et al., 2025), and the starting checkpoint. The value of each choice depends on the others: a dataset matters through the update that consumes it, while every run changes the evidence and checkpoints available next (Lambert et al., 2024; Guo et al., 2025; Qian et al., 2025). We therefore treat post-training as sequential search over joint recipes rather than independent parameter choices.

Language-model agents can now formulate hypotheses, execute experiments, and use observed outcomes to guide subsequent proposals (Lu et al., 2024; Yamada et al., 2025; Novikov et al., 2025), and recent systems extend this paradigm to data construction and post-training (Rank et al., 2026; Meng et al., 2026; Ma et al., 2026; Chen et al., 2026; Yu et al., 2026; NeoHorse Team et al., 2026). This feedback loop provides a process-level setting for *recursive self-improvement*, in which experimental evidence determines the next intervention on a target model. Existing formulations, however, either optimize data or parameters within a largely fixed training program (Zhang et al., 2023; Liu et al., 2024a; Meng et al., 2026; Ma et al., 2026), or permit open-ended pipeline changes that make controlled comparison difficult (Lu et al., 2024; Novikov et al., 2025; Rank et al., 2026). They therefore leave a central question unresolved: *can an autonomous researcher improve a model by jointly choosing the data, update procedure, optimization controls, and starting checkpoint across successive post-training experiments, and what evidence should determine its next intervention?*

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.

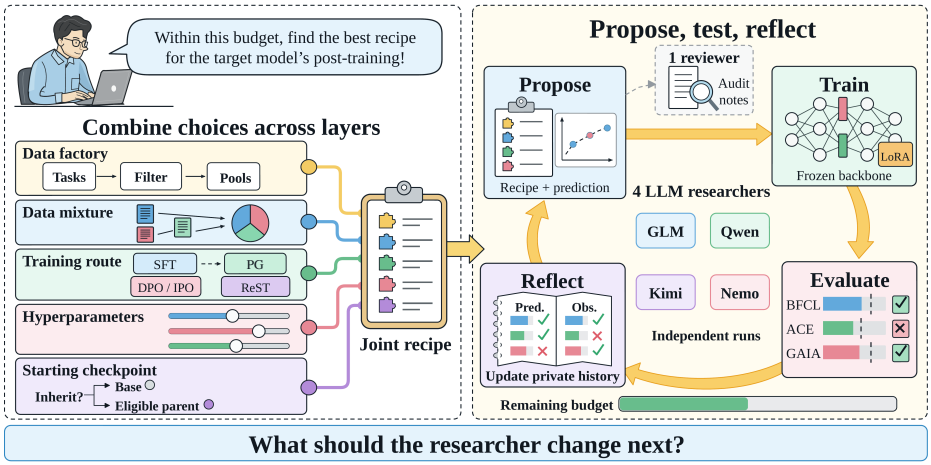

Figure 1: **SPOT: Stateful Post-Training Search.** Four LLM researchers independently propose joint recipes spanning data, updates, and checkpoint initialization under a shared budget. Shared execution and evaluation return outcomes and checkpoints to their private histories, closing the propose–train–evaluate–reflect loop.

We therefore introduce SPOT (**S**tateful **PO**st-**T**raining Search), which formulates autonomous post-training as sequential search over joint executable interventions. Instead of treating data, training, and initialization as separate tuning problems, each intervention specifies how supervision is constructed and mixed, which update procedure is used with its active optimization controls, and which eligible checkpoint is updated. At each round, a researcher uses its previous hypotheses, measured capability outcomes, available checkpoints, and remaining budget to propose the next intervention. SPOT validates and executes that proposal, evaluates the resulting model, and returns both the evidence and the new checkpoint to the research history. Figure 1 summarizes how this joint recipe is embedded in the resulting propose–train–evaluate–reflect loop. Although researchers may follow different search strategies, SPOT converts every proposal into a common structured recipe and executes it through the same training and evaluation pipeline. The resulting records make outcomes directly comparable, preserve checkpoint ancestry, and support analysis of both recipe quality and research behavior.

We instantiate SPOT to post-train Qwen3-8B (Qwen Team, 2025) with four LLM researchers: GLM-5.3 (Z.ai, 2026), Qwen3.8-27B (Qwen Team, 2026), Kimi-K2.6 (Moonshot AI, 2026), and Nemotron-3-Ultra-550B-A55B (NVIDIA, 2026). Each researcher searches the same joint recipe space for 12 rounds under a shared training budget and evaluation protocol. Candidates are evaluated on BFCL (Patil et al., 2025), ACEBench (Chen et al., 2025), and GAIA (Mialon et al., 2024). In this setting, every researcher discovers a checkpoint that improves over the initial Qwen3-8B model, with the best search-time composite score increasing from 0.4092 to 0.5152 on a 0–1 scale (a 25.9% relative gain). SPOT therefore discovers effective post-training recipes across heterogeneous autonomous researchers rather than in a single favorable run.

More importantly, SPOT's six search campaigns produce a large archive of complete training trajectories, aligning each proposed recipe with its realized update, step-level telemetry, evaluation outcomes, and checkpoint lineage. Together with controlled ablations, these trajectories support a three-question diagnostic hierarchy for deciding what to change next. **Does the sampled data still provide usable learning contrast?** Some runs contain both correct and incorrect responses, yet zero reward variation prevents any sampled group from entering the backward pass. A retrospective three-step probe relates the retained-group fraction (the proportion of sampled prompt groups used for the update) to subsequent exact-call gains (Section 5.1). **Does the surviving contrast reward the intended behavior?** Exact calls and name-only calls (correct function names but checker-rejected arguments) can receive identical credit. Across 92 paired trajectories, mean reward over the final third of training is nearly unrelated to the selected BFCL aggregate R^* (Spearman $\rho_S = 0.1023$). **Can the current recipe space express the required repair?** Of 92 unique matched trajectories, 77 remain below the base irrelevance score; recovery is therefore not reliable across the explored controls. Together, these questions distinguish when to continue optimization from when to change sampling, redesign the objective, or expand the intervention space.

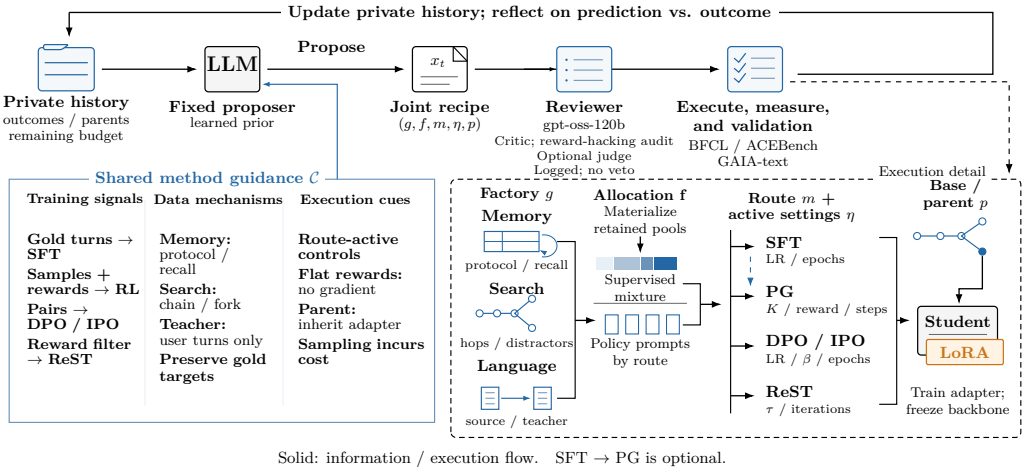

Figure 2: **Method overview.** A fixed LLM proposer combines its learned prior with private experimental history to propose the next joint recipe (g, f, m, η, p) under shared guidance $\mathcal{C}$. The recipe selects the data factory, allocation policy, training route and settings, and base or parent model. Execution constructs supervision from memory, search, or language mechanisms and trains a LoRA adapter using SFT, PG, DPO/IPO, or ReST while freezing the backbone. Outcomes from BFCL, ACEBench, and GAIA-text return to the private history for the next proposal. Solid arrows show information or execution flow; the dashed SFT-to-PG transition is optional.

Table 1: **Joint intervention space in SPOT.** Each proposal combines four coupled decision families; the selected update route activates only its compatible controls. Exact ranges and conditional rules appear in Appendix A.

Decision	Searchable choices	Controls
Data factory	Memory protocols; response endings and distractors; search graphs; teacher rewriting; format filters; task sources	What supervision is constructed
Data mixture	Shares of refusal, multi-turn, agentic, memory, search, and general data; rollout-prompt composition	Which capabilities receive it
Update	SFT, DPO/IPO, ReST, grouped policy gradient (PG), or SFT→PG; route-active learning rate, duration, sampling, reward, and adapter controls	How the signal changes the model
Initialization	Shared base with a fresh adapter, or an eligible checkpoint from the researcher's private campaign	Whether search restarts or inherits

2 METHOD

SPOT organizes autonomous post-training as a stateful loop over joint, executable interventions. As Figure 2 summarizes, each proposal is validated and executed, and its realized operation, outcomes, and checkpoint inform the next decision; we define the intervention space and feedback loop below.

Our grouped policy-gradient route (PG; logged as `grpo`) uses reward-weighted completion log-likelihoods, with group-normalized advantages by default, but no old-policy importance ratio, PPO clipping, or reference-policy KL term.

2.1 STATEFUL JOINT INTERVENTIONS

For researcher a at decision t, the state is $S_t^{(a)} = (H_t^{(a)}, \mathcal{A}_t^{(a)}, \mathbf{b}_t^{(a)})$, where $H_t^{(a)}$ is its private record of hypotheses, recipes, execution status, and measured outcomes; $\mathcal{A}_t^{(a)}$ contains its eligible checkpoints; and $\mathbf{b}_t^{(a)}$ is the remaining budget. Given this state and shared method guidance $\mathcal{C}$, the researcher returns a hypothesis q_t and a joint intervention

$$x_t = (g_t, \mathbf{f}_t, m_t, \eta_t, p_t). \tag{1}$$

Table 1 summarizes how each coordinate of x_t maps to an experimental decision. The data factory g_t chooses what supervision is constructed; the mixture $\mathbf{f}_t$ decides which capabilities receive it; m_t and its active controls η_t determine how the signal updates the model; and p_t chooses whether to restart from the shared base or inherit a measured checkpoint. These choices are coupled rather than independent: the update route determines which data and controls are active, mixture variables obey joint constraints, and completed experiments enlarge the set of eligible parents. Appendices A and A.3 give the complete schema, conditional ranges, and route-specific training details. Figure 3(a) maps LLM-proposed recipes and their scores into a landscape, with arrows indicating checkpoint inheritance.

2.2 Validated execution and stateful feedback

The submitted recipe is validated for required fields, legal bounds, route-specific consistency, and parent eligibility. Invalid proposals are returned for revision. Before dispatch, an advisory `gpt-oss-120b` reviewer provides configuration critiques and reward-hacking risk labels; an optional judge can label predicted score changes. These annotations are logged with the proposal and neither edit nor veto valid recipes under execute-all selection. The execution stage in Figure 2 resolves active fields, constructs and filters data, initializes from p_t, and applies the selected update under a fixed student, budget, and evaluation protocol. Suppressing the researcher superscript, one proposal-level transition is

$$S_t \xrightarrow{\pi_a(\cdot|\mathcal{C},S_t)} (q_t, x_t) \xrightarrow{\Phi,\, T,\, E} (\widetilde{x}_t, \theta_t, y_t) \xrightarrow{\text{append}} S_{t+1}, \tag{2}$$

where Φ validates and resolves the proposal, T executes training, and E evaluates the candidate. The realized intervention $\widetilde{x}_t$ records the parent, retained data, and settings actually used; θ_t is the resulting checkpoint and y_t its measured capability vector.

Crucially, SPOT stores x_t and $\widetilde{x}_t$ separately, together with execution status, training telemetry, evaluation outcomes, and checkpoint ancestry. This makes it possible to distinguish a poor intervention from a proposal whose intended learning signal was not actually instantiated. The resulting evidence is appended to H_{t+1}, and an eligible θ_t joins $\mathcal{A}_{t+1}$. A later proposal may cite an earlier experiment as evidence while initializing from a different eligible checkpoint; the former motivates the change, whereas the latter supplies its weights. Selecting the shared base is a *restart*, and selecting a prior checkpoint is *inheritance*. Researchers maintain independent histories, but all proposals pass through the same validation, execution, and evaluation interface, making their trajectories comparable and supporting the analyses in Sections 3 and 5. Appendix B.3 details execution and comparison audits.

3 Experiments and Results

3.1 Evaluation setting

Search campaigns. We instantiate SPOT to post-train Qwen3-8B (Qwen Team, 2025) with LoRA (Hu et al., 2021). The student learns tool use: selecting functions and arguments, memory operations, chained lookups, and multi-turn interactions. Training sources include synthetic key–value, vector, and recursive-summary memory tasks; chain and fork lookup tasks; and ToolACE or TicketAPI multi-turn examples, alongside single-turn, refusal, and long-horizon examples. Teacher rewriting changes eligible user-facing wording while preserving designated facts and targets.

Four independent twelve-round campaigns use GLM-5.3 (Z.ai, 2026), Qwen3.8-27B (Qwen Team, 2026), Kimi-K2.6 (Moonshot AI, 2026), and Nemotron-3-Ultra-550B-A55B (NVIDIA, 2026) as researchers. Each searches the joint intervention space in Table 1 under a common student, training budget, and evaluation protocol. Optuna TPE (Optuna, 2026) and random search provide classical references. In total, the six campaigns produce 141 fully evaluated candidate checkpoints: 97 from LLM-guided search, 21 from TPE, and 23 from random search. Appendix B.2 explains two-slot asynchronous scheduling and candidate accounting; Appendix B.3 details comparison boundaries.

Evaluation. We evaluate every completed checkpoint on selected BFCL tool-use categories (Patil et al., 2025), ACEBench normal and special (Chen et al., 2025), and GAIA-text (Mialon et al., 2024). The BFCL aggregate is $R^* = 0.10N + 0.10L + 0.10I + 0.30M + 0.40A$, where the five components measure non-live function calling, live function calling, irrelevance handling, multi-turn interaction,

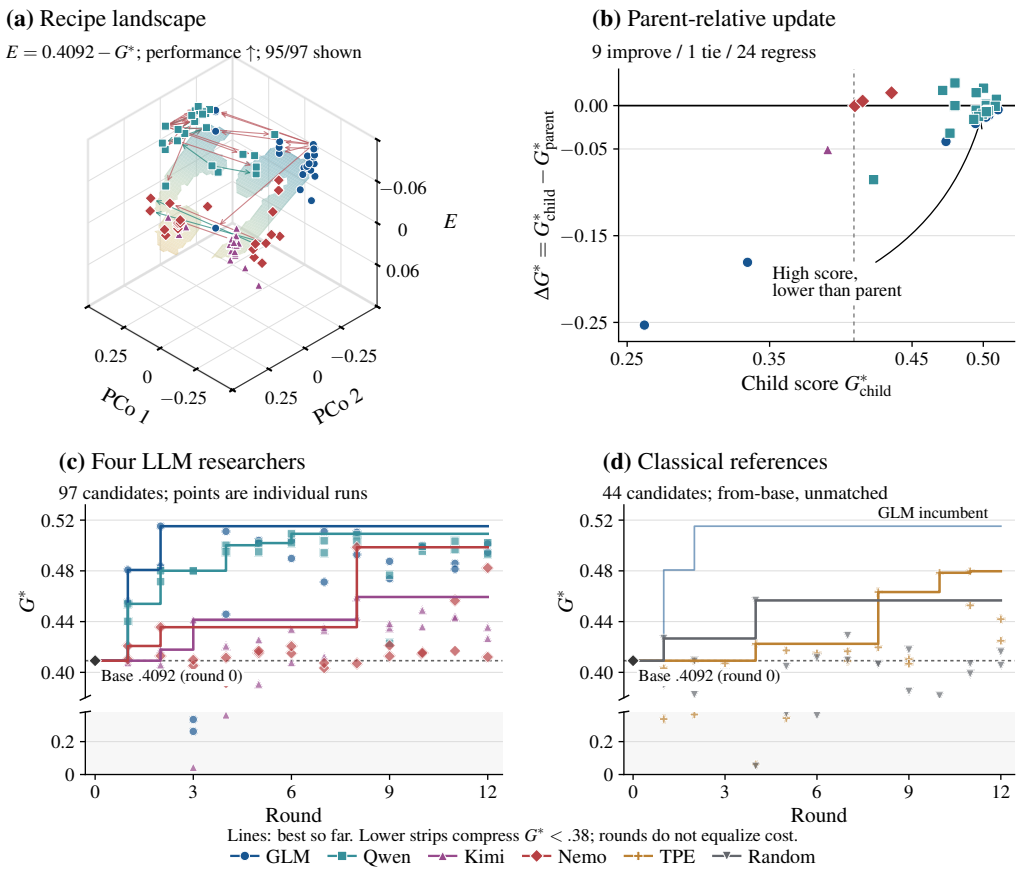

Figure 3: **Recipe quality, update value, and search progress.** (a) Score-independent recipe coordinates, measured $E = 0.4092 - G^*$ on a reversed axis, and parent-to-child arrows. The window shows 95/97 candidates; the pale fitted surface is descriptive. (b) All 34 scored parent-relative updates. (c) All 97 scored LLM candidates and four thin incumbent curves. (d) All 21 TPE and 23 random-search candidates, with GLM's incumbent for reference. Round 0 marks base ($G^* = 0.4092$). Lower strips compress scores below 0.38, including the two outcomes outside (a). Missing scores are not zero; rounds do not equalize cost.

and agentic memory/search, respectively. They aggregate 11 underlying category results; multi-turn and agentic performance receive 70% of the weight. ACEBench normal tests standard tool calling (250 items), special tests imperfect or unsupported requests (150), and GAIA-text is our 103-item text-only slice. The search-time utility is

$$G^* = 0.50R^* + 0.25s_{\mathrm{ACE},n} + 0.15s_{\mathrm{ACE},s} + 0.10s_{\mathrm{GAIA}}. \tag{3}$$

The coefficients are fixed, manually specified study-utility weights rather than statistically optimized values: BFCL receives half of the mass, while the remaining half is divided across ACEBench normal, ACEBench special, and GAIA-text as generalization measures. The shared base scores $G^* = 0.4092$. Because all four components are available during search, G^* measures search-time selection utility rather than held-out generalization. Appendix B gives the benchmark slices and scoring details, and Appendix A.1 gives the proposer prompts.

3.2 SPOT Discovers Strong Post-Training Recipes

SPOT delivers a large and broad gain. Across all four LLM researchers, SPOT discovers a checkpoint above the shared base: GLM reaches $G^* = 0.5152$, Qwen 0.5092, Nemo 0.4986, and Kimi 0.4594. The strongest checkpoint raises utility by 0.1060 on the 0–1 scale, a 25.9% relative gain, and exceeds the observed TPE and random-search peaks of 0.4797 and 0.4567. Moreover, the selected checkpoint from every LLM campaign improves over the base on R^*, ACEBench

normal, ACEBench special, and GAIA-text, so the aggregate gain does not conceal a decrease in any measured component. Figure 3(d) provides the classical references; because those campaigns use from-base initialization and are not cost-matched, they contextualize the achieved scores rather than isolate proposer quality. Appendix C.4 details the construction and interpretation of the landscape in Figure 3(a).

Success recurs across researchers and search stages. Figure 3(c) reports all 97 LLM-guided candidates rather than only the selected winners. All four researchers produce checkpoints above the shared base, but their incumbents improve at different stages: GLM finds its strongest result early, whereas Qwen and Nemo improve later.

Absolute quality and update value are different. Figure 3(b) evaluates all 34 scored parent-relative updates: 9 improve over their parent, 1 ties, and 24 regress. A child can therefore remain competitive in absolute score while failing to improve its initialization, so stateful search must track both quantities when deciding whether to extend a lineage.

Researchers adopt distinct search strategies. Figure 4 characterizes how the four researchers search under the same Full interface. Figure 4a shows that recipe-space coverage expands at different rates, revealing substantial variation in exploration coverage. Figure 4b shows distinct method preferences: GLM concentrates on SFT+PG, Kimi on PG, Nemo on ReST, while Qwen distributes its attempts more broadly. Figure 4c reveals an equally large difference in checkpoint reuse: Qwen inherits 22 of 24 candidates, compared with only 1 of 25 for Kimi. Together, the panels show that SPOT supports distinct stateful search strategies rather than prescribing a fixed recipe.

Taken together, the selected results and full archive show that SPOT discovers effective recipes across heterogeneous LLM research policies and multiple regions of the intervention space. Section 4 next asks how the interface shapes that search; Appendices D and C provide the candidate and trajectory breakdowns, and Appendix F contains the complete 150-record candidate register.

4 Interface Ablations

We ablate two choices in the SPOT interface: proposal scope and auxiliary annotations. *Full* permits joint changes to data, training, and initialization; *One-component* restricts each proposal to one recipe component; and *No-annotation* retains the full intervention space but removes the external critic, reward-hacking auditor, and optional judge. Figure 5 compares the realized searches. The proposal-scope comparison uses completed twelve-round campaigns from GLM, Qwen, Kimi, and Nemo. These records characterize realized searches, not independent replications or an intrinsic ranking of proposer models.

Joint proposal scope helps, but not uniformly. Under Full, a researcher can coordinate changes to the data factory, data mixture, update route and controls, and starting checkpoint. One-component proposals lower the best score from 0.5152 to 0.4993 for GLM and from 0.5092 to 0.5055 for Qwen. The difference is not confined to a single winner: their Full campaigns produce 18 of 48 candidates at or above 0.50, compared with only 1 of 49 under the restriction. Nemo and Kimi reverse the endpoint comparison: restriction raises their peaks from 0.4986 to 0.5060 and from 0.4594 to 0.4841, respectively. Thus, broader proposal scope is not monotonically beneficial. Joint changes can expose productive interactions, whereas a narrower scope can make local refinement more effective for a research policy that does not exploit the additional choices.

Recipe reuse does not imply checkpoint inheritance. Under Full, Qwen inherited checkpoints in 22 of 24 candidates, reaching lineage depth seven. Under One-component, all 96 scored candidates across the four researchers restarted from the shared base, although inheritance remained allowed. Nemo also shifted from ReST (12/24 candidates under Full) to SFT (19/24 under restriction). After its round-2 winner, all 20 later scored proposals in the restricted campaign cited that recipe, but every run restarted from the base and none improved on the winner. Nemo reused the recipe as a reference without extending its checkpoint lineage. These records show changes in update choice and checkpoint reuse that endpoint scores alone miss. The ablations describe realized search behavior rather than compute-matched causal effects; Appendices C and D.3 provide the audits.

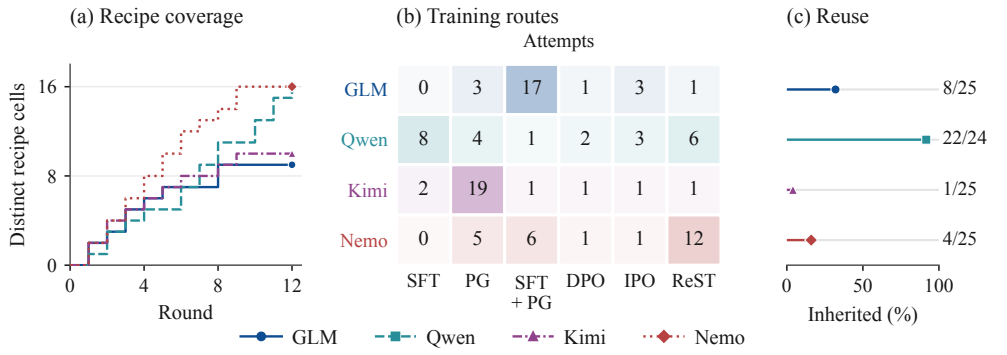

Figure 4: **Realized search behavior under the Full interface.** (a) Growth in occupied recipe cells, (b) attempted update routes, and (c) checkpoint-inheritance rates across the four LLM researchers. A recipe cell projects six discrete coordinates: update method, initialization type, active memory graph, active search graph, teacher, and eligible multi-turn source; it excludes continuous controls and specific parent identity.

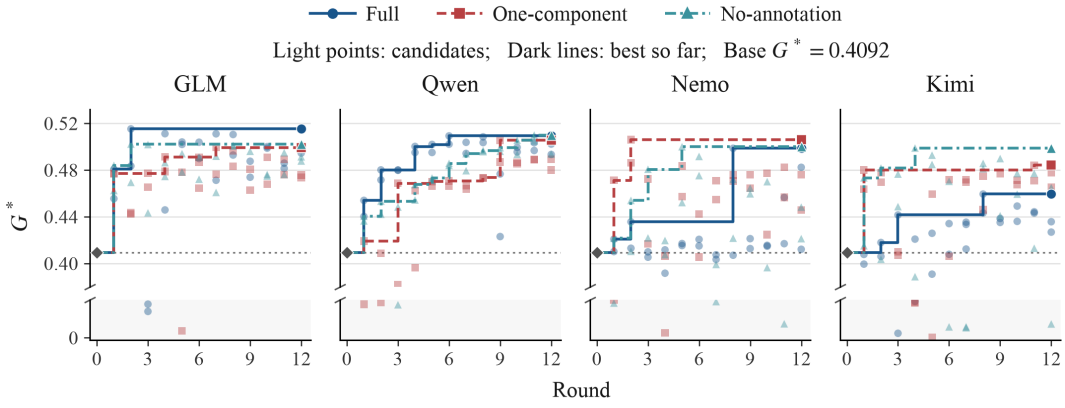

Figure 5: **Search under interface ablations.** Lines show base-inclusive incumbents and points show scored candidates in the displayed round order. Rounds are not compute-matched. Appendix C.3 specifies the analysis roster.

Researchers also improve the student without auxiliary annotations. All four No-annotation researchers produce checkpoints above the shared base across 94 measured candidates. Their private outcome history and reflection remain active, and annotations do not veto valid recipes under execute-all selection. These unmatched searches do not isolate the causal contribution of annotations. Section 5 examines diagnostic training evidence.

5 Diagnosing Post-Training Progress

Endpoint scores show which checkpoint performed well, but they do not explain why training improved or what should change next. We use the aligned training records to ask three questions in order: *does the batch provide a usable learning signal*, *does that signal reward the intended behavior*, and *can the current intervention space express the required repair?* The answer determines whether the next experiment should continue training, change reward or sampling, or expand the search space. These records aggregate evidence across trajectories rather than elevate a single run.

All three diagnoses require execution fidelity: the intervention described by the researcher must be the intervention consumed by training. SPOT records the proposed and realized operations separately, while Appendix A.5 and Appendix E provide the detailed audits and trajectory summaries.

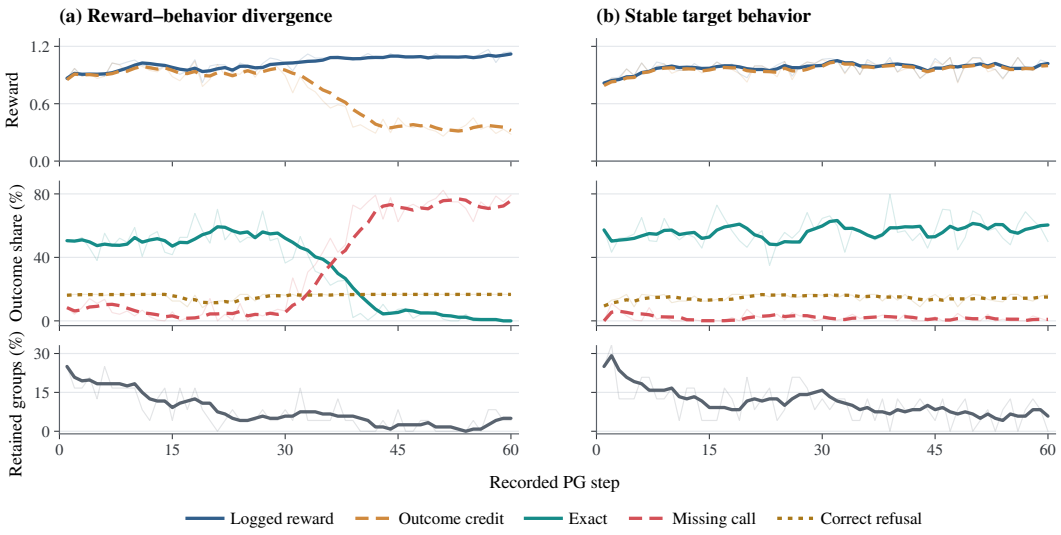

Figure 6: **Reward can improve while target behavior deteriorates.** In the divergent trajectory (left), reward rises as exact calls collapse and missing calls take over; the comparator (right) does not show this failure. Light lines show individual steps and dark lines summarize the local trend. Outcome credit is logged reward minus reconstructed format credit; an added format reward of 1 for eligible malformed outputs allows total reward to exceed 1.

5.1 Does the batch provide a usable learning signal?

Some batches in the archive contained both correct and incorrect calls yet contributed no policy loss: the reward assigned both behaviors the same value. Grouped policy-gradient (PG) updates need reward differences within a group to distinguish sampled responses. Repeating batches without this contrast does not provide a learning signal (Yu et al., 2025). We report the retained-group fraction as `groups_kept/groups`, where `groups` counts sampled prompt groups.

In a retrospective probe of 95 complete traces, we examined retention over the first three steps. Later exact-call gains exceeded .02 in 15/24 high-retention traces ($\geq .22$), 3/13 low-retention traces ($< .08$), and 29/58 intermediate traces. The probe covers 2.5–12% of recorded steps; this is a step-count fraction, not a measured compute-cost fraction. These associations do not validate a classifier or stopping rule. Appendix E.1 gives the probe protocol.

A batch's error rate alone does not tell us whether it provides usable contrast. An error-heavy slice can be uninformative when its responses receive equal rewards, whereas a slice with fewer errors can be more useful if the reward separates the target behaviors. Data selection should account for both the remaining errors and how often sampled groups provide this contrast. Before extending a run, check that the reward distinguishes the errors of interest and that sampled tasks produce such groups. Table 11 and Figure 10(a) give concrete zero-retention cases. If contrast is absent, change reward or sampling rather than extend the training horizon.

5.2 Does the signal reward the intended behavior?

A usable learning signal can still optimize the wrong behavior. Figure 6 shows a trajectory in which logged reward continues to rise while exact tool calls disappear and missing calls take over. The comparator does not show the same collapse. Both traces are from No-annotation and are not matched interventions (Appendix E.1). Rising reward is therefore not sufficient evidence that the target behavior is improving.

Aggregate exact-call share also mixes behavior quality with how often call-bearing cases occur. A decline in the overall share can therefore coexist with improving exactness within call-related outcomes. We first separate this composition effect from conditional behavior; only a within-behavior collapse, as in Figure 6, supports changing the objective.

Component-level accounting explains the divergence: an auxiliary format term rewards malformed tool-call outputs even as credit for the primary behavior falls. The broader lesson is not tied to this particular term. Whenever the training objective is a proxy, its components must be checked against the behavior the experiment is intended to improve. If reward and target behavior diverge, the next step is to redesign the objective before allocating more training. Appendix A.5 contains the full reconstruction.

5.3 Can the current search space express the repair?

Persistent failure under varied controls suggests a missing intervention, not insufficient optimization. Most matched PG candidates remain below the base irrelevance score (Appendix E). Figure 9 shows the selection consequence: requiring both GAIA and irrelevance retention makes four search arms fall back to the base checkpoint. This evidence concerns the explored space; it does not imply that the behavior is unlearnable.

Earlier-stage success also does not validate the next stage. Table 13 shows that a targeted irrelevance repair improves its target while reducing ACE-special and aggregate utility; similarly, stable SFT can lead to different RL outcomes once sampling and reward change. Stage transitions therefore require a fresh check of error composition, usable contrast, and objective alignment.

The next experiment should therefore introduce a qualitatively different intervention, such as a new task slice, supervision source, or objective decomposition. Continuing to adjust the same controls is unlikely to resolve a failure that persists across the explored space. Each step presumes execution fidelity and a valid learning signal.

6 Related Work

Human-designed post-training recipes. Post-training couples data, supervision, optimization, and the model state to update. Tülu 3 develops such a multistage recipe, while DAPO and Dr. GRPO show how sampling and objective design alter the usable learning signal (Lambert et al., 2024; Yu et al., 2025; Liu et al., 2025b). We study how an LLM chooses among these coupled interventions and what the resulting training archive reveals.

LLM-guided optimization and experimentation. LLAMBO and AgentHPO use language-model priors and experimental history for hyperparameter search; population-based training jointly adapts weights and hyperparameters; and The AI Scientist and Gome (*Reasoning as Gradient*) automate broader experimental revision (Liu et al., 2024a; 2025a; Jaderberg et al., 2017; Lu et al., 2024; Zhang et al., 2026). SPOT instead constrains proposals to executable post-training recipes and records the realized intervention, outcome, and lineage needed to audit later decisions.

Autonomous post-training research. PostTrainBench and RSIBench-Data evaluate post-training and data-centric research agents; LaMDAgent and AutoTrainess expose training actions, model reuse, and structured records; and TREX and SAGE combine LLM researchers with tree search (Rank et al., 2026; Meng et al., 2026; Yano et al., 2025; Yu et al., 2026; Ma et al., 2026; Zhao et al., 2026). SPOT gives the LLM direct control over conditional data construction, supervision mixtures, update procedures, active hyperparameters, and checkpoint initialization through a shared executor. It separates experimental references from checkpoint ancestry and uses aligned execution records to analyze recipe discovery and training progress. Within the LLM arms, no external tree-search or Bayesian-optimization policy selects the next experiment.

7 Conclusion

SPOT makes joint post-training choices and their consequences inspectable. The observed campaigns find checkpoints with higher search-time utility, but capability retention and parent-relative updates give a different account of progress from the best composite score alone. Training records further expose reward mappings that discard behaviorally useful distinctions. In a divergent trajectory, code-linked accounting traces the reward increase to format credit while primary call outcomes deteriorate. These findings identify concrete questions for subsequent experiments: whether the intended intervention was consumed, whether its reward distinguishes target behavior, and whether apparent progress survives capability and ancestry checks. The study is limited to one student in a bounded tool-use setting and non-replicated adaptive searches, without an untouched post-selection test or campaign-level uncertainty estimates. Incomplete historical provenance and unmatched controls constrain attribution, while diagnostic interventions and stopping rules remain prospectively unvalidated.

AI USE STATEMENT

Generative AI assisted literature triage, hypothesis formulation, search-space design, synthetic training-data generation, software implementation, experiment orchestration (the controller is itself an LLM), interpretation of training logs and results, figure preparation, and prose drafting. The human authors are responsible for checking all citations, derivations, code, artifacts, and claims. No private free-form model chain-of-thought is treated as an experimental artifact or published as evidence; the recorded research objects are structured configurations and their explicit hypothesis fields, validation results, executed training steps, and frozen evaluation scores.

ETHICS STATEMENT

SPOT studies automated post-training in a bounded tool-use setting. Such automation can amplify errors or undesirable behavior when a training reward is an imperfect proxy: this study documents reward–behavior divergence and loss of irrelevance handling. Benchmark gains should therefore not be interpreted as evidence of safe deployment. Applying the framework beyond this setting requires task-specific safety and bias evaluation, limits on tool permissions, and human oversight. Use or redistribution of model weights, benchmark data, and generated supervision must respect their licenses and applicable privacy requirements. The reported search and execution audits expose observed failures and provenance gaps; they do not certify the resulting models as safe or fair.

REPRODUCIBILITY STATEMENT

Section 2 defines the stateful search interface, and Section 3 specifies the experimental setting and selection utility. Appendix A documents proposer inputs, data construction, conditional training controls, and execution semantics; Appendix B details benchmark slices, scoring, candidate accounting, and comparison boundaries. Appendices E and F provide the diagnostic analysis protocol and candidate-level register. These descriptions support inspection and reanalysis of the reported archive. Exact reruns also depend on historical prompt and provider settings and on artifacts whose availability is qualified in Appendix B.2; we do not claim bitwise reproducibility of adaptive campaigns.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024. URL https://arxiv.org/abs/2402.14740.

Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Yuefeng Huang, et al. ACEBench: Who wins the match point in tool usage? *arXiv preprint arXiv:2501.12851*, 2025. URL https://arxiv.org/abs/2501.12851.

Wanyi Chen, Xiao Yang, Xu Yang, Tianming Sha, Qizheng Li, Zhuo Wang, Bowen Xian, Fang Kong, Weiqing Liu, and Jiang Bian. Agent^2 RL-Bench: Can LLM agents engineer agentic RL post-training? *arXiv preprint arXiv:2604.10547*, 2026. URL https://arxiv.org/abs/2604.10547.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4447–4455. PMLR, 2024. URL https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023. URL https://arxiv.org/abs/2308.08998.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through

reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017. URL https://arxiv.org/abs/1711.09846.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, et al. Tülu 3: Pushing frontiers in open language model post-training, 2024. URL https://arxiv.org/abs/2411.15124.

Siyi Liu, Chen Gao, and Yong Li. AgentHPO: Large language model agent for hyper-parameter optimization. In *Conference on Parsimony and Learning*, volume 280 of *Proceedings of Machine Learning Research*, pp. 1146–1169. PMLR, 2025a. URL https://proceedings.mlr.press/v280/liu25c.html.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/84b8d9fcb4e262fcd429544697e1e720-Abstract-Conference.html.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, et al. ToolACE: Winning the points of LLM function calling, 2024b. URL https://arxiv.org/abs/2409.00920.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective, 2025b. URL https://arxiv.org/abs/2503.20783.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, et al. APIGen: Automated pipeline for generating verifiable and diverse function-calling datasets, 2024c. URL https://arxiv.org/abs/2406.18518.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards fully automated open-ended scientific discovery, 2024. URL https://arxiv.org/abs/2408.06292.

Zerun Ma, Guoqiang Wang, Xinchen Xie, Yicheng Chen, He Du, Bowen Li, Yanan Sun, Wenran Liu, Kai Chen, and Yining Li. TREX: Automating LLM fine-tuning via agent-driven tree-based exploration. *arXiv preprint arXiv:2604.14116*, 2026. URL https://arxiv.org/abs/2604.14116.

Fanqing Meng, Lingxiao Du, Qiguang Chen, Ziqi Zhao, Haocheng Lu, Mengkang Hu, and Michael Qizhe Shieh. RSIBench-Data: Benchmarking data-centric research for recursive self-improvement. *arXiv preprint arXiv:2607.25886*, 2026. URL https://arxiv.org/abs/2607.25886.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A Benchmark for general AI assistants. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25ae35b5b1738d80f1f03a8713e405ec-Paper-Conference.pdf.

Moonshot AI. Kimi-K2.6 model card. Official model card, 2026. URL https://huggingface.co/moonshotai/Kimi-K2.6. Accessed September 25, 2026.

NeoHorse Team et al. NeoHorse-1: Towards recursive self-improvement via agentic post-training with routing harness. *arXiv preprint arXiv:2609.08183*, 2026. URL https://arxiv.org/abs/2609.08183.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131.

NVIDIA. Nemotron 3 Ultra: Open, efficient mixture-of-experts hybrid Mamba-Transformer model for agentic reasoning. Technical report, 2026. URL https://research.nvidia.com/labs/nemotron/files/NVIDIA-Nemotron-3-Ultra-Technical-Report.pdf.

Optuna. *TPESampler: Optuna 4.9.0 API Documentation*, 2026. URL https://optuna.readthedocs.io/en/v4.9.0/reference/samplers/generated/optuna.samplers.TPESampler.html. Accessed September 25, 2026.

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. PMLR, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

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

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, 2025. URL https://arxiv.org/abs/2504.13958.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3.8 model repository. Official model repository, 2026. URL https://github.com/QwenLM/Qwen3.8. Accessed September 25, 2026.

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. URL https://arxiv.org/abs/2305.18290.

Ben Rank, Hardik Bhatnagar, Ameya Prabhu, Shira Eisenberg, Karina Nguyen, Matthias Bethge, and Maksym Andriushchenko. PostTrainBench: Can LLM agents automate LLM post-training? *arXiv preprint arXiv:2603.08640*, 2026. URL https://arxiv.org/abs/2603.08640.

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, 2024. URL https://arxiv.org/abs/2402.03300.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-level automated scientific discovery via agentic tree search, 2025. URL https://arxiv.org/abs/2504.08066.

Taro Yano, Yoichi Ishibashi, and Masafumi Oyamada. LaMDAgent: An autonomous framework for post-training pipeline optimization via LLM agents, 2025. URL https://arxiv.org/abs/2505.21963.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, et al. DAPO: An open-source LLM reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Zhaojian Yu, Penghao Yin, Shuzheng Gao, Shilin He, Kai Cai, and Xiao-Ping Zhang. AutoTrainess: Teaching language models to improve language models autonomously. *arXiv preprint arXiv:2606.31551*, 2026. URL https://arxiv.org/abs/2606.31551.

Z.ai. GLM-5.3 model repository. Official model repository, 2026. URL https://github.com/zai-org/GLM-5. Accessed September 25, 2026.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488. Curran Associates, Inc., 2022. doi: 10.52202/068431-1126. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.

Michael R. Zhang, Nishkrit Desai, Juhan Bae, Jonathan Lorraine, and Jimmy Ba. Using large language models for hyperparameter optimization. *arXiv preprint arXiv:2312.04528*, 2023. URL https://arxiv.org/abs/2312.04528.

Yifei Zhang, Xu Yang, Xiao Yang, Bowen Xian, Qizheng Li, Shikai Fang, Jingyuan Li, Jian Wang, Mingrui Xu, Weiqing Liu, and Jiang Bian. Reasoning as gradient: Scaling MLE agents beyond tree search, 2026. URL https://arxiv.org/abs/2603.01692.

Haoran Zhao, Wei Du, Dingwen Yang, Jixuan Huang, Junlin Shang, Lingyong Fang, Ya Guo, Tao Gui, Qi Zhang, and Xuanjing Huang. Strategy accumulation and guided execution for automated LLM fine-tuning, 2026. URL https://arxiv.org/abs/2609.22257.

A EXECUTABLE SEARCH INTERFACE

This appendix describes the frozen execution implementation used as the audit reference, rather than earlier runner variants. The interface distinguishes proposed, active, derived, and ignored controls. Shared field names alone do not establish matched search support: defaults, bounds, normalization, conditional activation, and parent permissions also matter.

A.1 PROMPT AND RESEARCH STATE

Each proposer receives the student description, JSON schema, method/factory cards, training goal, evaluation definitions, base measurements, its available outcome history, and eligible checkpoints. Other arms' histories and external champion checkpoints are withheld. An experiment cited as evidence need not supply the initialization weights: references motivate a proposal, whereas parents identify the adapter actually loaded. The hosted proposer weights remain fixed; student-training `self_critique` is a separate mechanism.

Proposals pair a configuration with a hypothesis and expected response. Early round-3 prompts request confidence, evidence, expected upside, expected failure mode, and `what_would_falsify`. Later versions shorten this to confidence, evidence identifiers, and a falsification bucket. The recorded round-3 gate requires confidence, evidence, and `what_would_falsify`; confidence is a low/medium/high label, not a calibrated probability. From round 3, a deterministic `EXPERIENCE` block summarizes the arm's underperforming methods and prediction–outcome mismatches. The LLM interprets this evidence when proposing its next experiment. Prompt wording, retry guidance, and requirements evolved; the histories do not represent one invariant prompt. Provider settings, prompt versions, and feedback availability are therefore part of the reproduction contract.

One private campaign.

1 Initialize private history and the shared base.
2 At an available slot, render the schema, conditional space, limits, available scores, capability buckets, references, and parents.
3 Request a JSON recipe, hypothesis, and expected response.
4 Normalize and check configuration and parent eligibility; request revision if invalid.
5 Materialize route-specific data, load the base or eligible parent, and train with active controls.
6 Record execution status and available telemetry; score completed checkpoints and compute G^* when the vector is complete.
7 Return available outcomes and eligible checkpoints to private history.
8 Report the best complete-vector checkpoint, with base fallback.

Round 1 has no prior candidate outcomes or transfer/G^* feedback; later calls use only measurements then available, not the completed vectors reconstructed for final analysis. Schema checks establish field presence and legal references, not the truth of a hypothesis.

A.2 DATA CONSTRUCTION AND MIXTURES

Table 2 separates allocation across protocols, interaction structure, and language variation. Second recall asks for a different field after a recall-ending episode; it does not permit arbitrary memory-operation sequences. Vector memory has no replacement operation, so update probability is inactive on that slice. Fork search uses two one-hop lookups. TicketAPI covers login, creation, and retrieval with bounded recovery cases. Larger pools do not expand these structures.

Mixture demand. For source pools $P_q(g)$ and shuffled sampling operator U, the nominal supervised mixture allocates five shares to refusal, multi-turn, long-horizon, memory, and search, with a single-turn remainder:

$$D(x) = \sum_{q=1}^{5} f_q U(P_q(g)) + (1 - \textstyle\sum_q f_q) U(P_0(g)), \qquad f_q \in [0,1], \quad \textstyle\sum_q f_q \le 1. \tag{4}$$

Integer sampling, finite pools, filtering, and sampling with replacement can change realized allocation or repeat examples.

Table 2: Factory controls and activation conditions. Integer ranges include their endpoints. Memory/search controls matter when their supervised or agentic-rollout share is positive; ReST also activates these pools.

Family	Allowed values	Condition or coupling
Memory protocol	KV / vector / recursive-summary shares in $[0, 1]$	Sum to one; validator tolerance 0.01
Memory structure	`linear`, `second_recall`; distractors 0–4	Second recall changes recall-ending episodes; seven fields are fixed
Memory endings	Abstain $[0, .5]$, update $[0, .5]$, end-irrelevant $[0, .4]$	Sum $\leq .8$; recall receives the remainder; vector has no replace operation
Search structure	`linear`, `fork`; hops $1, 2, 3$	Hops apply to linear chains; fork has two one-hop lookups
Search content	Entities 4–16; distractors 0–5	Synthetic facts and answers
Filtering	`all`, `official`	Checker filtering changes retained-pool composition
Language	Template or paraphrase; teacher none / Qwen / DeepSeek	Paraphrase requires a teacher; rewrite share $[0, 1]$
Teacher content	None or `recall_q`; user layer only	Changes user questions, with designated facts and targets retained
Multi-turn source	ToolACE or TicketAPI state machine	State-machine pool size 200–1500; recovery share $[0, .4]$
Other sources	Irrelevance: ToolACE / factory; auxiliary fill: frozen / verified synthesis	Replacement affects the corresponding mixture component

Let f_m, f_s denote supervised memory/search shares and $a = \texttt{agentic_prompt_frac}$. With nominal size $N = 9000$ and repetition target $\kappa_{\text{rep}} = 1$, the sizing helper uses $e = 1$ for PG-only shots and the configured/default epoch count otherwise:

$$u_m = Ne(f_m + a/2)/\kappa_{\text{rep}}, \qquad u_s = Ne(f_s + a/2)/\kappa_{\text{rep}}.$$

ReST floors these demands at 512 and 256, respectively. For positive demand, or for ReST, generated sizes are

$$n_m = \text{clip}(\lceil u_m \rceil, 200, 3000), \qquad n_s = \text{clip}(\lceil u_s \rceil, 200, 2000).$$

Zero demand outside ReST retains cache defaults $n_m = 3000, n_s = 1500$; these do not imply consumption. Pool sizes and the seven-field memory setting are derived/pinned. This demand heuristic includes rollout needs even when supervised shares vanish, but does not count exact trainer consumption. Clipping and filtering can increase repetition; the helper's ReST epoch field does not set the trainer's supervised passes. Generated, retained, unique, and consumed quantities are therefore distinct.

A.3 Conditional training and rewards

Table 3 gives route-specific controls. Parent loading preserves adapter architecture and weights while restarting the optimizer. SFT+PG executes both stages. Preference training and ReST generate policy-dependent data, so equal trials, epochs, or nominal examples do not imply equal cost.

SFT minimizes assistant-token cross-entropy. DPO/IPO pair gold assistant turns with current-policy errors (Rafailov et al., 2023; Gheshlaghi Azar et al., 2024). Their reference is the shot's initial policy, including any parent adapter:

$$h_\theta = [\log \pi_\theta(y^+|q) - \log \pi_\theta(y^-|q)] - [\log \pi_{\text{ref}}(y^+|q) - \log \pi_{\text{ref}}(y^-|q)].$$

The losses are $-\log \sigma(\beta h_\theta)$ and $(h_\theta - 1/(2\beta))^2$, respectively; response reduction and masks follow the executor version. The shared beta field has different effects in the two objectives. ReST alternates generation, absolute reward-threshold retention, and supervised fitting (Gulcehre et al., 2023); reward scale affects its retained set.

The grouped policy-gradient trainer (PG; historical route key `grpo`) assigns rewards $r_1, \ldots, r_K$ to completions of a prompt. Its default REINFORCE-style objective uses group-normalized advantages:

$$A_i = \frac{r_i - \bar{r}}{\sigma_r + \epsilon}, \qquad \mathcal{L}_{\text{PG}} = -\frac{1}{N_{\text{tok}}} \sum_i \sum_{u \in y_i} A_i \log \pi_\theta(y_{i,u} \mid q_i, y_{i,<u}).$$

Table 3: Training controls and fixed caps for SFT, grouped policy gradient (PG), SFT+PG, DPO, IPO, and ReST. Bounds define the interface, not optimal values or equal compute. LR sampling parameterization also affects search support.

Scope	Searchable values	Execution qualification
Fresh LoRA	Rank $8, 16, 32, 64$; alpha/rank $.5, 1, 2$; dropout $[0, .1]$; targets all-linear / attention / MLP	Parent loading retains the parent's adapter architecture
SFT stage	LR $[10^{-5}, 2\times10^{-3}]$; epochs $1, 2, 3$; linear / cosine / constant schedule; warmup 0–50	Assistant-token loss; 8192 token limit
Grouped policy gradient	LR $[10^{-6}, 5\times10^{-5}]$; steps 20–120; group $4, 8, 12$; temperature $[.7, 1.1]$	3072 default sequence limit; 256 new-token limit; 24 prompts per step
Rollout composition	Refusal share $[0, .6]$; agentic share $[0, .5]$	Sum $\leq .9$; agentic demand feeds both synthetic pools
Grouped reward / advantage	Five weights $[-1, 2]$; five advantage modes; optional self-critique	Surrogate-reward controls, not evaluation weights
DPO / IPO	LR $[10^{-6}, 5\times10^{-5}]$; beta $[.01, .5]$; epochs $1, 2$	At most 512 retained preference pairs in the recorded dispatch
ReST	LR $[10^{-5}, 2\times10^{-3}]$; iterations 1–4; completions $4, 6, 8$; threshold $[.5, 1]$	Prompt cap 512; iterations $\times K \times 512 \leq 10000$

Nearly constant-reward groups are skipped. This implementation uses group-relative advantages (Shao et al., 2024), without an old-policy importance ratio, PPO clipping, value head, or reference-policy KL term. Other modes retain positive normalized advantages, weight group maxima, use leave-one-out centering, or change sequence normalization. The `gspo` option is not a clipped GSPO objective.

Training reward uses official BFCL checker components on synthetic expectations, with partial-credit and wire-format terms:

$$\mathcal{R}_{\text{train}}(y, e; w) = \max\{0, v(y, e; w) + w_{\text{fmt}} m(y)\}.$$

The false-call refusal branch is zero; v selects exact-call, name-only, no-call-refusal, or answer-match credit, and m flags malformed tool-call markup. Positive w_{fmt} therefore rewards malformed output. Reward is not evaluation accuracy. Global positive scaling approximately cancels in normalized and positive-only advantages, but epsilon, variance cutoffs, retention thresholds, and self-critique triggers preclude exact pipeline-wide scale invariance. Raw reward magnitudes are not comparable across recipes.

A.4 Execution and audit qualifications

Normalization fills defaults and removes selected derived keys before checks of choices, bounds, mixtures, and parents. Validity rules define executable actions; resource caps bound cost; design restrictions fix the student, tools, method menu, and prohibition on free-form code edits. Admission does not establish semantic truth, security isolation, or independent validation. Factory-side answer-overlap exclusions can inspect benchmark answer strings; this access is distinct from proposer visibility.

The historical deployment used two workers with four A800-80GB GPUs each, initially two GPU slots per proposer; released slots later served classical controls. Each recorded round nominally allocated two candidate proposals per researcher. Slots were refilled asynchronously from the researcher's then-available history, including completed outcomes from the same round; the two slots therefore did not necessarily share a fixed round-start history. Retained failures and refill records can make the realized count exceed two configurations in a round; the complete accounting is in Appendix B.2. Missing parents, failed training, incomplete evaluation, and completed low scores retain distinct statuses. Failures consume resources but are not zero-score checkpoint evaluations. Parent identifiers must resolve to loaded adapter bytes; a "damaged" audit label was not itself a parent veto. Ancestor execution is counted once in search expenditure and remains part of cumulative lineage cost.

Several implementation qualifications constrain interpretation. Positive format weights permit malformed calls to outscore well-formed ones. Generic no-tool-call preference construction can misclassify a factual answer as a refusal target. Parent architecture overrides new rank/target proposals; fork disables hops; ReST accepts schedule, warmup, and epoch fields that its dispatch does not apply. Numeric coercion can normalize non-finite input unexpectedly, without evidence that reported trials used it. These are historical issues, not repairs performed by revising the manuscript.

The cache's truncated configuration digest omits code, seed, and data identity; repeated proposals may reuse one realization. Some runtime records are incomplete and can include queueing/evaluation, so they do not establish matched GPU cost. A complete attempt manifest would bind raw/effective configurations, visible history, prompt and code versions, parent/data digests, seeds, generated/retained/unique/consumed quantities, status, scores, and resources. Not every historical row contains these fields. Repair helpers rewrite ledgers with backups, so the historical ledger is not strictly append-only; exports must preserve amendments and distinguish historical from corrected executor versions.

A.5 Execution-impact audit

This audit links the frozen execution implementation to archived configurations, training logs, evaluator summaries, and available ledger backups. It covers the six selected checkpoints and four diagnostic cases. The source package includes code excerpts, source/log hashes, per-step calculations, and the audit script in `data/revision_20260925/audit/`. This is an audit of existing runs; no training, reward repair, or independent evaluation was performed. The archive does not attest the remote runtime source hash for every run, so source-conditioned inferences assume that the frozen implementation and recorded weights match execution.

Score reconciliation. For all six selected checkpoints, archived BFCL category maps and transfer summaries match their ledger entries. Reapplying the recorded formulas reproduces their R^* and G^* at recorded precision. Available backups contain no different scored R^* or G^* for these checkpoints. This checks evaluator-summary arithmetic and agreement among available records; it is not per-item rescoring, evidence that all earlier records were immutable, or confirmation on unseen data.

Positive format credit was exercised under the frozen semantics. The reward parser marks malformed tool-call markup, including invalid JSON or missing function names. On eligible branches, a positive `w_format` adds credit for this failure. False-call and reward-error branches remain hard zero. Among all scored configurations, 37 of 97 main, 4 of 21 TPE, and 5 of 23 Random candidates both contain a policy-gradient stage and admit a positive format weight. These are configuration-exposure counts, not direct measurements of malformed output.

The malformed flag is not printed in the training log. Its aggregate reward contribution can nevertheless be reconstructed in the seven audited policy-gradient traces (Table 4). Let N_t be all outcome-kind counts, including `crit_*` revisions, and let B_t be their verdict-tier reward sum. This sum uses the configured exact, name-only, answer, and correct-refusal weights after stripping the revision prefix. The audited traces have nonnegative tier weights, positive format weight f, and no unaccounted reward kinds, so clipping introduces no unknown correction. The integer count M_t of bonus-receiving malformed completions satisfies

$$N_t(r_t - .0005) \;\le\; B_t + fM_t \;\le\; N_t(r_t + .0005), \qquad 0 \le M_t \le N_t^{\text{eligible}}, \tag{5}$$

where r_t is the logged mean rounded to three decimals and N_t^{eligible} excludes hard-zero branches. Enumerating the feasible integers, with a numerical tolerance at rounding boundaries, gives one solution at each of 440/440 steps. An independent implementation re-read the raw rows and reproduced these solutions with decimal arithmetic. The count excludes malformed outputs on hard-zero branches and does not identify individual outputs or gradient-contributing groups.

For `6f022f362b58`, the first versus final five steps have mean logged rewards .9084 and 1.1198, while the reconstructed outcome-credit component falls from .8963 to .3202 and the format component rises from .0121 to .7995. Their sum agrees with logged reward within rounding. At step 60, with no revisions, 192 outputs include 152 missing calls; 165 bonus applications are inferred. At least 125 missing calls must therefore have received format credit. The comparison trace has much smaller first/final format components, .0091 and .0218. This accounts for rewards on the observed outputs,

Table 4: Source-conditioned reconstruction of positive format-credit applications. Counts include initial completions and student revisions; they are inferred from rounded rewards and verdict counts, not directly logged format flags.

Role	Candidate	Steps	f	Inferred M	All outputs
GLM selected	`0a02d4fcb0f4`	40	.3	122	7,680
Kimi selected	`1318c27e04f7`	80	1.5	392	16,674
Nemo selected	`5ea7197d3d46`	80	1	325	15,360
Divergence	`6f022f362b58`	60	1	3,987	11,608
Comparison	`3ddd3c7873b4`	60	1	174	11,638
Critique case	`7d407ef3adca`	60	1	4,684	11,604
Zero-kept case	`b61baaa2d2de`	60	1	93	11,520

not a trajectory that would result from corrected training. It identifies an implementation- associated failure in this case rather than a general law about extending correctly specified reinforcement learning.

The student-side critique scheduler skips a group when its maximum reward is at least .99. Format credit can consequently change whether critique is requested. Falling critique counts cannot be interpreted independently as evidence that the critic improved or failed. Reviewer annotations in the proposal history are a separate mechanism.

Accepted and consumed controls. Qwen's selected `995a40213227` and its five recorded ancestors use only SFT and ReST; they do not enter the proposal-controlled PG format branch. The selected TPE and Random checkpoints use from-base SFT. The Qwen ReST recipe declares cosine scheduling and warmup, but the frozen dispatch does not pass these fields to its trainer, which uses constant-rate AdamW. Similarly, `epochs` does not control ReST's supervised passes, although it can affect pool sizing upstream. Its log records two retained sets of 2,558/3,072 and 2,712/3,072 completions, with no stepwise loss curve. These records cannot support a claim about the effect of the declared schedule or warmup. The audit lists accepted controls for all 28 main/control ReST records. Parent loading also retains the existing adapter architecture; new rank/target declarations need not be new architecture interventions.

Cache and ledger scope. The 97 main, 21 TPE, and 23 Random scored records have 141 distinct candidate identifiers, with no repeated scored identifier in these ledgers. The main archive contains 12 repeated generator identities covering 28 records. This demonstrates shared generated-pool identities, not duplicated checkpoint training or 141 independent realizations. Missing full data, seed, and runtime digests prevent a stronger identity claim. Ledger repair helpers can rewrite records while saving backups; the ledger is therefore not strictly append-only. The six-checkpoint reconciliation above bounds the observed impact on reported selected scores without certifying the complete mutation history. These findings leave measured score arithmetic intact while narrowing execution and causal claims.

B Evaluation protocol and comparison audit

B.1 Benchmark slices and utility

For the in-loop BFCL subset, the recorded capability vector and aggregate are

$$r(x) = (N, L, I, M, A), \qquad R^*(x) = 0.10N + 0.10L + 0.10I + 0.30M + 0.40A,$$

where N averages Python-simple and multiple, L averages live-simple and live-multiple, I is non-live irrelevance, and M is multi-turn-base. The agentic component is

$$A = \tfrac{1}{2}\left[\mathrm{mean}(\mathrm{KV}, \mathrm{vector}, \mathrm{rec_sum}) + \mathrm{mean}(\mathrm{web_base}, \mathrm{web_no_snippet})\right].$$

These are official checker components on selected categories (Patil et al., 2025), not the full leaderboard aggregate. Complete comparisons require all eleven category results. Evaluation uses a fixed Qwen non-thinking wrapper.

The search-time utility uses the fixed coefficients in Equation 3. BFCL receives half of the composite weight, with the other half allocated to ACEBench normal, ACEBench special, and GAIA-text. These values are shared by all campaigns and are not search variables. ACEBench normal and special test different tool-use contracts (Chen et al., 2025), on 250 and 150 items, respectively. GAIA-text uses 103 items, a separate driver, and the official answer scorer (Mialon et al., 2024). The coefficients define the study's utility, rather than statistically optimized weights; component vectors remain necessary for interpreting the aggregate. ACEBench and GAIA are transfer benchmarks relative to student training, but their `xfer` scores are computed in the loop, enter G^*, and return in proposer history. None of these components is an untouched post-selection test.

Because all four components enter proposal history, G^* is the declared search objective rather than a held-out generalization measure. An idealized budget-B objective evaluates the incumbent

$$\widehat{\theta}_{\pi,B} = \theta_{\arg\max_{i:C_i \le B} G^*(\theta_i)},$$

where C_i is cumulative search expenditure through candidate i, including failed attempts and proposal overhead. The reported endpoint comparisons use the completed archives, not measured equal-B estimates: incomplete cost records prevent matching this full expenditure across arms. An untouched data, task, or model axis would address the separate question of post-selection transfer; it must be evaluated only after freezing $\widehat{\theta}_{\pi,B}$ and must not be used to reselect among candidates.

B.2 Archive provenance and candidate accounting

Reconciliation on 14 September 2026 checked 978 files in four experiment directories. Each matched its version-controlled copy in the frozen main-campaign snapshot. This is the archive reconciliation for the main comparison, distinct from the 25 September ablation export (Appendix C.3). The main ledger has identical bytes across these two snapshots. For the later export, file-level digests identify the actual exported data; the recorded source version alone does not establish that every exported ledger matches its version-controlled copy. The four LLM arms share a ledger with private proposer visibility; TPE and Random have separate ledgers. Random's two official round-12 sidecar candidates also occur in its main ledger and are deduplicated by configuration ID. Adapters, pools, working directories, driver logs, and Optuna databases are not thereby claimed as uploaded artifacts; the export records source hashes and per-file publication status.

There are 147 scheduled records: 99 LLM, 24 TPE, and 24 Random. Three additional Random round-12 infrastructure records bring the total to 150 unique configurations. Of these, 141 have complete official vectors: 97 LLM, 21 TPE, and 23 Random. The nine unscored records comprise two LLM training failures, three TPE generation/evaluation failures, one Random evaluation timeout, and three Random infrastructure records. Missing evaluations are not zero scores, and non-official timeout sidecars are not promoted to official results. Completed low-scoring checkpoints remain included. The nominal two-proposal schedule is not an exact per-round count: GLM round 3 and Nemo round 9 each retain a training failure alongside two scored candidates. Kimi round 7 retains three scored candidates: after one candidate completed and its companion encountered a teacher-generation timeout, the recovered slots received two further proposals. All three scored records remain in the register; the extra candidate is not the shared base checkpoint. All 99 LLM configurations and previously archived vectors reconciled without changes; the supplied figure sources agree with all 72 arm-round maxima exactly and all 96 component means/medians within four-decimal rounding.

B.3 Classical search support and sampler phase

The archived TPE uses Optuna 4.9.0 with `multivariate=True` and `group=True`, modeling dependencies within decomposed conditional subspaces (Optuna, 2026). Startup ends after ten completed observations. Database timestamps identify ten scored startup draws and eleven adaptive draws, the latter beginning in round 7. Failed or repeated events make trial IDs insufficient for this classification. Database objectives reconcile with all 44 official classical vectors. Random's two residual `RUNNING` states are superseded by abort tombstones; its official round-12 results come from the sidecar database. The databases are local-only; the manuscript includes the sanitized audit.

Both classical arms fix `init_from` to null, excluding the LLMs' growing parent action set. Their sampler also omits `sft_lr` for SFT+PG, although it supplies the field for SFT/ReST. All six scored classical SFT+PG runs therefore use the validator default 4.73×10^{-4} in their SFT stage, with a

Table 5: Candidate accounting and available executor-slot time. Time coverage counts records with the field, not necessarily scored records. Random's three extra infrastructure records are outside the scheduled column.

Arm	Scheduled	Official	Time records	Slot minutes
GLM	25	24	20	8967.8
Qwen	24	24	20	4567.3
Kimi	25	25	22	7886.4
Nemo	25	24	22	9060.1
TPE	24	21	22	9620.6
Random	24	23	23	9368.8

sampled second-stage rate; GLM's sixteen scored scratch SFT+PG runs use 10^{-4}. Source branches, effective configurations, and absent database entries agree. This is restricted support, not an inferred optimizer preference. The comparison does not isolate proposer quality without matched parent access, active controls, and costs.

B.4 Resource coverage and confirmation boundary

The `gpu_minutes` field measures elapsed executor-slot time through generation, waits, training, and evaluation, not GPU busy time (Table 5). Unequal missingness and overlapping slots prevent a matched GPU-hour comparison. The proposal-cost summary, dated 6 September, covers partial rounds, estimates input tokens using a common tokenizer, and lacks output tokens; it is not a complete campaign bill.

Independent confirmation remains future work. Such a protocol would freeze the checkpoint selected by search-time G^* at a declared budget, then evaluate data or task/model axes excluded from selection. Its export must identify checkpoint IDs, selection time, excluded items, exposure policy, and outcome files. Repeated searches would compare these frozen choices, not select the largest independent-test score across candidates. The historical measurements reported here do not establish that additional confirmation.

C Search behavior and checkpoint lineage

C.1 Active recipe cells and sensitivity

We analyze all 99 non-base LLM execution records in the frozen ledger, including two failed attempts. A recipe cell uses six coordinates: *training method, scratch/inherited initialization, active memory graph, active search graph, active teacher, and route-eligible multi-turn source*. Parent identity and continuous controls remain outside this projection.

Memory/search graph controls are active when their supervised share is positive, a policy stage has positive agentic-prompt share, or ReST consumes the generated prompts. Teacher identity counts only for an eligible paraphrase or recall-rewriting route. Multi-turn source counts only for a mixed-data route with positive `multiturn_frac`; pure policy-gradient and ReST paths do not consume that override. A field's presence in an accepted proposal is insufficient. These activation rules follow the inspected source snapshot, which does not independently attest every historical worker executable. Filtering and subsampling can further reduce exposure; realized diversity requires example-level consumption records.

Table 6 changes the projection and roster. The four-coordinate view omits teacher and multi-turn source; the eight-coordinate view adds active `teacher_content` and `gen_keep`. Qwen and Nemo exchange order, but both visit more cells than GLM/Kimi under these checks. For cell counts c_j among N attempts, uniform subsampling to k attempts gives expected coverage

$$\mathbb{E}[C(k)] = \sum_j \left[1 - \binom{N - c_j}{k} \Big/ \binom{N}{k}\right],$$

Table 6: Visited cells under alternative explicit projections and subsets of the same frozen ledger. These are counts, not percentages of all possible executable recipes.

Projection or subset	GLM	Qwen	Kimi	Nemo
Four coordinates	7	10	8	13
Six coordinates	9	16	10	16
Eight coordinates	10	17	10	16
Without initialization	9	15	10	13
Factory coordinates only	6	10	6	12
Six, scored candidates only	9	16	10	15
Six, rounds 1–11 only	9	15	10	16

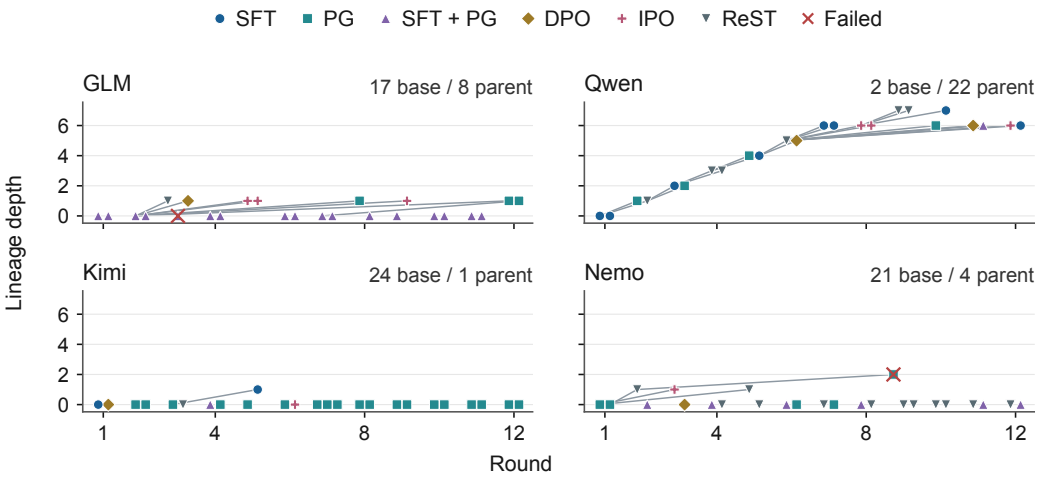

Figure 7: **Checkpoint ancestry across the four private campaigns.** Edges identify requested initialization parents; red crosses mark failed attempts. Depth zero means a scratch attempt. A requested edge does not certify completed training. Same-round offsets improve legibility only. PG denotes grouped policy gradient (historical key `grpo`). All 99 nodes and 35 requested parent edges are shown.

with impossible numerator combinations set to zero. At $k = 22$, the values are 8.40/15.15/9.04/14.54 for GLM/Qwen/Kimi/Nemo. This is descriptive rarefaction, not a temporal prefix or an equal-budget experiment.

C.2 Inheritance, progress, and remaining choices

Figure 7 preserves the actual ancestry graph. Qwen inherits in 22/24 attempts, compared with 8/25 for GLM, 1/25 for Kimi, and 4/25 for Nemo. Qwen's successful maximum depth is seven; the other campaigns reach one. Nemo's attempted depth-two child fails. All 22 Qwen inherited attempts switch training method, as do all 34 scored inheritance edges across arms. The shared method cards explicitly discuss losses from same-method stacking and successful method-switch lineages; these choices do not isolate an unaided pretraining preference.

Qwen's peak path is

`77013bf0` → `a3ec6de9` → `524566d9` → `0ebb8ab8` → `a6d55136` → `995a4021`.

This path has depth five and reaches $G^* = .5092$; later depth-seven branches do not improve it. Its peak has nine direct children, and child `63f5e04c` has three. Seven of ten early parent-relative changes are numerically positive, versus none of twelve later changes. Small differences and dependent descendants prevent interpreting these counts as reliable improvement probabilities.

Table 7 separates score plateaus from further exploration. New cells can reveal poor routes; changes within an occupied cell can still repair a capability. The observed allocation does not establish

Table 7: Post-peak allocation and late novelty. Attempts after the peak exclude all candidates in the peak's round. Cells are the six-coordinate projection. No arm records a later higher G^*.

Arm	Peak round	Attempts after peak round	New cells R9–12	Best G^*
GLM	2	21	0	.5152
Qwen	6	12	5	.5092
Kimi	8	8	1	.4594
Nemo	8	9	2	.4986

Table 8: Measured ablation cohorts. Each entry gives the number of scored candidates and their peak G^*. Full and One-component have measured results through round 12 for every arm. The final column gives the No-annotation displayed round range. These are descriptive archive cohorts, not independent repetitions or matched-compute experiments.

	Full		One-component		No-annotation		
Arm	n	Peak	n	Peak	n	Peak	Display rounds
GLM	24	.5152	25	.4993	23	.5022	1–12
Qwen	24	.5092	24	.5055	24	.5099	1–12
Nemo	24	.4986	24	.5060	23	.4999	1–12
Kimi	25	.4594	23	.4841	24	.4988	1–12

a stopping rule. A prospective test would branch from a common checkpoint and history into continuation, restart, and method-switch conditions at the same remaining budget. To distinguish guidance from reflection, method cards and parent options should remain fixed while outcome history is retained, withheld, or permuted. These are follow-up designs, not experiments reported here.

C.3 Ablation analysis cohorts and allocation

The original ablation analysis uses the 25 September export from the main, one-component, and no-annotation ledgers. The export preserves SHA-256 digests of the actual ledger bytes. This later ablation export is distinct from the 14 September main-campaign snapshot in Appendix B.2; their main-ledger bytes agree. File-level digests identify the actual exported data independently of the recorded source version. This snapshot is preserved in `data/appendix_20260925/ablation_snapshot/`. The current analysis roster updates only the Kimi No-annotation series with the 26 September measured-score export in `data/ablation_update_20260926/`; all other measured series are unchanged. The two ablation conditions contain 202 assembled execution records: 190 have complete measured scores and 12 do not. Together with the 97 scored Full candidates, the comparison covers 287 measured candidates. Table 8 reports the twelve measured series. Provisional scores are excluded, and missing evaluations never enter incumbent curves.

One-component has 102 recorded candidates, including six without complete scores. The updated No-annotation roster has 100 assembled records, including six without complete scores. Execution identity uses both configuration and seed rather than configuration ID alone. Displayed round indices organize the plotted records; they do not establish matched histories or matched computational budgets.

Checkpoint inheritance remains legal under One-component but is unused in all 96 scored candidates. Qwen changes from 22/24 inherited Full candidates to none under the restriction. Nemo's allocation shifts from 12 ReST candidates among 24 scored Full candidates to 19 SFT candidates among 24 restricted candidates. All 20 of its restricted candidates after round 2 cite the winning recipe `1a5fe8a4`, start from base, and score below its .5060 peak. GLM and Qwen jointly produce 18/48 Full candidates at or above .50, versus 1/49 restricted candidates. These differences describe how scope and realized allocation co-vary; they do not isolate the cause of the score differences. No-annotation retains private outcome history and reflection, so removing annotations does not remove the feedback loop.

C.4 Recipe landscape: construction and interpretation

Score-independent construction. Figure 3(a) shows the archived recipe projection. Its dissimilarity gives equal weight to seven blocks: memory, search, language/source, supervision allocation, training-update controls, reward shaping, and initialization. Components are equally weighted within each block. Numerical differences are divided by their observed active ranges, with learning rates first transformed by $\log_{10}$; categorical differences use zero/one mismatch and mixtures use total variation. Two inactive values contribute zero, while an active–inactive pair contributes one. Principal-coordinate analysis supplies the two leading positive-eigenvalue axes, retaining 65.94% of positive inertia. Scores, proposer labels, rounds, and exact parent identities do not determine the horizontal coordinates. These axes summarize represented recipe differences, not individual hyperparameters or the fraction of the search space explored.

Surface and display. Measured height is $E = 0.4092 - G^*$, displayed on a reversed axis so better outcomes appear higher. The window shows 95 of 97 scored candidates and 33 of 34 scored ancestry edges; arrows connect actual parents to children. The two off-window outcomes remain in Figure 3(c). The archived Gaussian radial-basis kernel-ridge fit uses all 97 scored outcomes, including the two outside the displayed $E \in [-0.12, 0.12]$ window. Four-fold validation keeps coincident projected positions together when choosing the bandwidth and regularization; the displayed fit uses lengthscale 0.5865 and ridge strength 1. The surface is shown only inside the observed convex hull and where the third distinct scored neighbor lies within the 90th percentile of observed leave-self-out third-neighbor distances. This geometry-based mask is not a confidence region. The source archive preserves the coordinates, scores, fitted grid, and mask; restoring the main-text panel changes none of them.

Interpretation and limits. Competitive checkpoints occur in several projected regions, and parent-to-child arrows traverse different neighborhoods. This complements the score and ancestry views in Figure 3(b–d): a high-scoring child need not improve its parent. Projected proximity does not imply identical recipes or outcomes; coincident positions in this archive differ by up to 0.1393 in measured G^*. The smooth surface summarizes the observed mixture of recipes and starting checkpoints. It was not used to select proposals, does not certify that an interpolated location is executable, and should not be read as a validated optimum or a causal effect of moving through recipe space.

D Capability profiles and checkpoint retention

D.1 Selected checkpoints and within-round winners

A campaign-selected checkpoint maximizes observed G^* over that campaign; a within-round winner maximizes it only among candidates in the same scheduling round. Neither permits assembling component-wise maxima from different checkpoints. Table 14 preserves all 150 records, including the nine unscored attempts, and marks the 72 within-round winners with stars. Missing evaluations remain missing rather than becoming zeros.

Similar composite scores can conceal different capability profiles. GLM's selected round-2 checkpoint has $(R^*, G^*, \text{ACE–N}, \text{ACE–S}, \text{GAIA}) = (.4694, .5152, .8080, .4133, .1650)$; Qwen's round-6 selection has $(.4451, .5092, .7800, .5333, .1165)$. Their composite gap is only .0060, yet Qwen is .1200 higher on ACE-special, while GLM is .0243 higher on R^* and .0485 higher on GAIA. Figure 8 decomposes each selected checkpoint's gain using the fixed utility weights. This is an arithmetic identity, not attribution to a recipe component. GLM's 10.60-point composite gain includes 4.12 points from BFCL and 5.799 from the two ACE components. Qwen's 10.00-point gain includes 6.899 points from ACE and only .194 from GAIA; Nemo's 8.94-point gain includes 6.699 from ACE. Thus, comparable aggregate improvements can place different demands on subsequent capability retention.

Selection and retention also differ within a campaign. Six of Qwen's eight within-round winners with $G^* \geq .49$ score below the GAIA base (.0971), although its campaign-selected checkpoint scores .1165. That selected gain amounts to roughly two of 103 GAIA items. No LLM campaign's G^*-selected checkpoint is also its GAIA-maximizing checkpoint. These observations concern selected snapshots, not replicated search means.

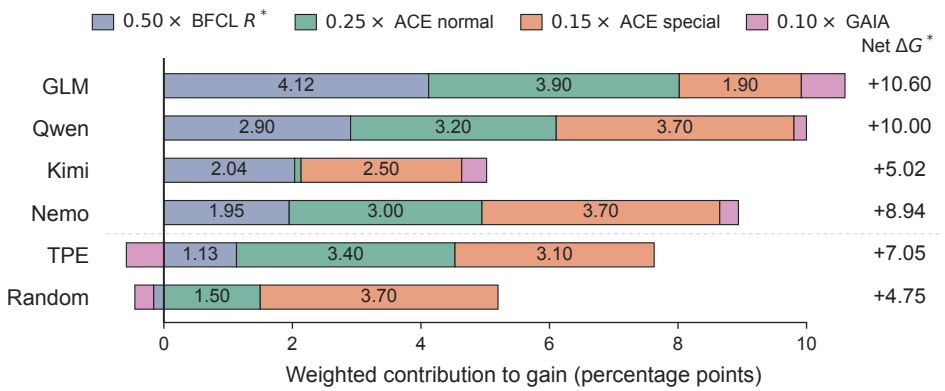

Figure 8: **Capability contributions to selected-checkpoint gains.** Signed, weighted differences from the shared base sum to the observed G^* gain up to rounding. Negative contributions extend left of zero; the right column reports net gains. The decomposition does not identify which training choice caused a change.

Table 9: Candidate means (upper block) and medians (lower block). All 141 scored candidates are included; n is the count within an arm. Component medians need not describe one checkpoint.

Arm	n	G^*	R^*	ACE-N	ACE-S	GAIA	Irr.	MT	Agentic
Mean									
Base	–	0.4092	0.3870	0.6520	0.2867	0.0971	0.8667	0.2400	0.1397
GLM	24	0.4748	0.4159	0.7587	0.4389	0.1141	0.7302	0.3398	0.1684
Qwen	24	0.4879	0.4332	0.7643	0.4822	0.0785	0.7325	0.3717	0.1813
Kimi	25	0.4100	0.3846	0.6261	0.3277	0.1204	0.7535	0.3056	0.1236
Nemo	24	0.4214	0.3995	0.6587	0.3181	0.0930	0.7943	0.3010	0.1425
TPE	21	0.3993	0.3699	0.6366	0.3127	0.0828	0.7663	0.2431	0.1303
Random	23	0.3877	0.3716	0.6061	0.2814	0.0819	0.7696	0.2593	0.1318
Median									
Base	–	0.4092	0.3870	0.6520	0.2867	0.0971	0.8667	0.2400	0.1397
GLM	24	0.4913	0.4304	0.7840	0.4567	0.1068	0.7500	0.3675	0.1895
Qwen	24	0.4952	0.4413	0.7760	0.4900	0.0728	0.7396	0.3850	0.1850
Kimi	25	0.4334	0.3977	0.6520	0.3467	0.1165	0.8042	0.3200	0.1285
Nemo	24	0.4148	0.3957	0.6520	0.3000	0.0922	0.8354	0.2775	0.1387
TPE	21	0.4150	0.3824	0.6560	0.3067	0.0874	0.8250	0.2500	0.1356
Random	23	0.4054	0.3838	0.6480	0.2867	0.0777	0.8500	0.2550	0.1408

D.2 Distributions over all scored candidates

Candidate summaries weight each of the 141 complete official vectors equally within its arm: 24/24/25/24 candidates for GLM/Qwen/Kimi/Nemo, and 21/23 for TPE/Random. Table 9 reports mean and median values for eight measurements under this common denominator. The scored ReST collapses at $G^* = .0440$ (Kimi), .0619 (TPE), and .0500 (Random) remain included. They are measured degradation, unlike the unscored infrastructure failures.

Peak discovery does not describe typical candidate quality. Qwen has higher mean and median G^* than GLM despite GLM's higher peak, and Qwen's minimum (.4230) exceeds the .4092 base. GLM, Kimi, and Nemo have 22/24, 18/25, and 19/24 candidates above base, respectively. Nemo's .4986 peak contrasts with a .4148 median. Qwen's mean/median GAIA (.0785/.0728) fall below base, whereas Kimi's (.1204/.1165) are highest among the LLM campaigns. All four have mean and median irrelevance below base; this does not imply every candidate regresses.

Descendants of a selected parent are dependent observations. These distributions therefore characterize the realized archive, not search reliability across independent campaigns. Moreover, component-

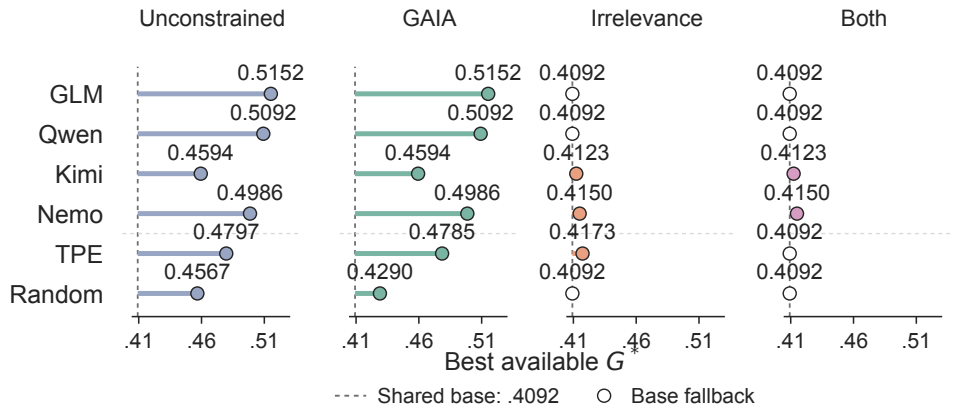

Figure 9: **Best available utility under post-hoc retention constraints.** Each panel reselects from the same scored archive, requiring the named capabilities to score at least as highly as the base. Open circles indicate base fallback; dashed lines mark the shared base $G^* = .4092$, and horizontal segments show gains over it. Small feasible gains remain descriptive search-time measurements.

wise medians need not describe any actual checkpoint, and their weighted sum need not equal median G^*.

D.3 Post-hoc retention sensitivity

For scored candidates $\mathcal{E}_a$, base x_0, and retrospectively chosen retained components $\mathcal{K}$, define

$$U_a(\mathcal{K}) = \max_{x \in \mathcal{E}_a \cup \{x_0\}} \{G^*(x) : s_k(x) \geq s_k(x_0) \text{ for every } k \in \mathcal{K}\}.$$

The base remains feasible. Figure 9 applies these constraints at recorded four-decimal precision; it does not rerun search with retention-aware feedback.

TPE's unconstrained round-11 selection has vector $(.4096, .4797, .7880, .4933, .0388)$. Requiring GAIA retention selects round 10, with $(.4216, .4785, .7600, .4533, .0971)$: a .0012 composite difference exchanges about six GAIA items for ACE performance. Random's unconstrained peak has irrelevance .2000 versus the base's .8667. Requiring both GAIA and irrelevance retention leaves four arms with base fallback. These tradeoffs motivate prespecified retention criteria and paired evaluation; the retrospective constraints provide neither an independent test nor an optimizer ranking.

E Training diagnostics and proposal case studies

E.1 Reading the reward–outcome traces

Figure 6 compares two complete, 60-step policy-gradient traces from `ma_v61_ablate_nocritic`: `6f022f362b58` and `3ddd3c7873b4`. Both are observed runs, not matched interventions. The source files are named `train_MA_<id>.log_s2` under that campaign's `analysis/train_logs/` directory. The source package preserves the 120 step records, source hashes, plotting code, and numerical checks in `data/revision_20260925/figures/`.

Fields and denominators. Each logged policy step reports mean reward, retained groups, critique count, advantage mode, and outcome-kind counts. Name-only denotes a correct function-name multiset with rejected arguments. The primary-outcome population includes exact, wrong, name-only, missing-call, false-call, answer-ok, answer-missing, and refusal-correct events; it excludes `crit_*` revisions. The displayed exact, missing-call, and refusal-correct shares divide by this same primary population. They are not conditional call accuracy or fixed-task accuracy. Logged mean reward includes both initial completions and critique revisions. The latter population difference is explicit rather than interpreted as a behavioral effect. Retained-group fraction is `groups_kept/groups`. Advantage mode is a configuration, not a changing learned state.

Table 10: Window annotations for Figure 6. Behavior shares use all primary outcomes; reward includes revisions. All windows contain five recorded steps and 960 primary completions.

Run	Window	Reward	Exact (%)	Missing (%)	Retained (%)
`6f022f362b58`	26–30	1.0236	52.08	5.31	5.83
	56–60	1.1198	0.00	75.73	5.00
`3ddd3c7873b4`	1–5	.8836	51.98	4.38	19.17
	56–60	1.0202	60.42	.94	5.83

Table 11: Selected zero-retention steps from main-campaign logs. Counts refer to outputs at the shown step; other outcome kinds are omitted here. The recorded reward configuration gives exact and name-only outputs equal base credit in these cases.

Run	Step	Exact	Name-only	Kept groups
`b61baaa2d2de`	25	124	28	0/24
`aee2a1381e5d`	62	102	34	0/24
`947cc3703729`	9	127	17	0/24

Pale lines preserve all recorded steps. Dark lines use trailing windows of five recorded steps: rewards are averaged and fractions pool the corresponding counts before division. The first four steps have raw traces only. There is no estimated causal onset or online alarm in the figure. Table 10 supplies exact window annotations; these annotate rather than replace the full trajectories.

The primary outcome composition changes sharply in the first case, while the comparison retains exact calls despite a similarly low final retained-group fraction. The configurations and accounting of reward terms are audited in Appendix A.5. These selected traces do not estimate failure prevalence or demonstrate that changing a reward or mixture would repair the model. SFT records, where available, contain five-step token-average loss blocks; a shorter final block is not a comparable endpoint. Terminal telemetry is never substituted for a curve.

Output diversity without useful reward contrast. Three inspected steps combine exact and name-only outputs while retaining no groups (Table 11). Each uses eight samples per group; the exact/name-only counts are not multiples of eight. Distinct output quality can therefore coexist with no retained group when the reward mapping fails to distinguish those outputs. These are selected execution examples, not a frequency estimate or evidence that all discarded groups would have been useful under another objective.

Training-trajectory analysis protocol. We freeze a roster from the campaign-assigned training-log locations of six v6.1 archives: the main campaign, the two interface ablations, the no-annotation continuation, TPE, and Random. These directories are not six independent replications. We require at least nine recorded PG steps, a complete sequence from step one to the declared horizon, and terminal telemetry. All retained traces also contain an adapter-save marker. Of 96 eligible files, two are byte-identical copies of `7d70f0e67827` in the no-annotation and continuation archives; we count this training run once, leaving 95 traces. Of these, 92 have a matched ledger R^*. Historical `pre_v61` files and unassigned `unmatched` files are outside this roster. We apply no outcome-based "ordinary-run," collapse, or reward-weight filter. The parser, roster, source hashes, and parsed numerical traces accompany the source in `data/submission_statistics/`.

At each step, exact and error shares divide the corresponding count by all non-critic outcome counts, including non-call outcomes. The main full-trace figure uses the same primary-outcome population, with pooled counts within its trailing windows. Logged mean reward includes any student-side critique revisions, whereas the outcome shares exclude `crit_*` counts. Window summaries average per-step quantities. The probe compares the first three steps with the final $\lfloor n/3 \rfloor$ steps: exact-share gain must exceed .02. The high-retention bin includes all early fractions at least .22 (including one); the low bin is below .08. The respective counts are 15/24 and 3/13; the intermediate bin contributes 29/58. All three bins use the 95-trace roster, not the 92 traces paired with R^*. The retained-group fraction is `groups_kept/groups`, where `groups` counts sampled prompt groups. The first three

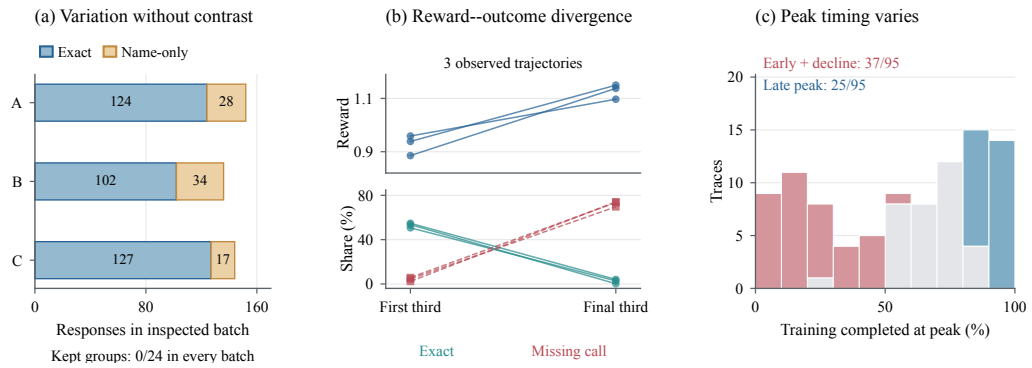

Figure 10: **Supplementary retrospective summaries.** (a) Three selected steps mix exact and name-only outputs but retain no groups. (b) First- and final-third means for three selected traces, with reward including revisions and outcome fractions using primary outputs. (c) Peak positions for all 95 complete distinct traces under the centered five-step definition above. The red and blue subsets contain 37 early peaks followed by a decline and 25 late peaks. Thresholds are retrospective and do not constitute validated stopping or continuation rules.

steps represent 2.5–12% of recorded horizons; this is a step fraction, not a measured wall-clock or compute-cost ratio.

Spearman correlations use average ranks for ties and final-third means, not terminal telemetry aggregates. On the 92 paired traces, correlations with R^* are $\rho_S = .1023$ for reward and .4879 for exact share. Their top-20 overlaps with the R^* top 20 are one and seven; none of these rankings has a tie crossing the selection boundary.

Peak analysis uses centered five-step exact-share means with shorter edge windows. Count ratios, window means, and peak ties use exact rational arithmetic. The earliest maximum defines peak position $(i + 1)/n$, without rounding. An early overrun requires a peak in the first half and a drop of at least .05 relative to the final-third mean of the smoothed series: 37/95 qualify. In 33 of these 37, raw mean reward is higher in the final third than in the first third; this does not imply monotonic or post-peak reward improvement. Another 25/95 peak in the final 15%. These are training-rollout diagnostics, not intermediate-checkpoint evaluations or validated stopping thresholds.

Additional archive summaries. Of 51 runs whose hard-error share decreases by more than .03 between the first and final third, 27 show a larger increase in name-only share than in exact share. Hard errors here sum wrong, missing-call, and false-call events. This composition change does not establish per-response transitions because the sampled batches change. Among 92 candidates with matched BFCL results, 77 fall below the base irrelevance score, three tie, and twelve exceed it. These counts describe this trace roster; the main capability-retention table instead uses the 141-candidate search archive. Neither roster estimates independent replication probabilities.

E.2 From available evidence to a proposed recipe

The five cases in Table 12 were selected retrospectively to expose initial composition, parent-based repair, recombination, and revision after a failed expectation. Candidate IDs resolve to Table 14. We paraphrase recorded hypothesis fields rather than publishing private free-form reasoning. These bundled comparisons do not attribute outcomes to individual controls.

Archived GLM round-3 and Kimi round-4 prompts contain an `EXPERIENCE` block and five requested hypothesis fields; later versions use shorter evidence/falsification tokens. Explicit same-round result citations can establish ordering that a round label alone cannot, because slots are refilled asynchronously. Initial proposals already use base measurements and authored method guidance; later ones also use own outcomes. Neither case isolates an unaided prior or the causal value of reflection. Admission and retry filtering further separate raw proposals from the terminal execution roster: rejection-code counts divided by executions are not proposer error probabilities.

Table 12: Selected links from an available observation to a proposed intervention and its measured outcome. These are bundled recipe comparisons, not causal estimates for individual controls.

Case	Evidence used in the proposal	Cross-layer proposal	Outcome relative to the expectation
GLM R1, `5088c115`	Shared base has weak memory/search buckets; no own trained history.	Generated protocol examples, memory/search mixture, SFT–PG sequence and rollout allocation.	$G^* = .4807$, but KV $= .0323$ is below the base .0968 and contradicts the positive KV bet.
Qwen R2, `a3ec6de9`	Own SFT parent has useful search performance and weak irrelevance/recursive-summary memory.	Continue the parent with ReST; increase memory share, emphasize recursive summaries, use factory refusal.	Recursive-summary memory and irrelevance rise; search and ACE-special fall. G^* : .4540 $\rightarrow$ .4801.
Kimi R4, `aee2a138`, `7a833133`	Different own recipes support KV recall and ACE-special; the hypothesis attempts to combine their strengths.	Alter factory/training options in a scratch PG recipe, or compose SFT and PG stages.	The PG variant recovers KV but misses its ACE target. The fused recipe falls to $G^* = .3625$.
GLM R3, `18e64b3b`	DPO child `8bc58b23` repaired target buckets while damaging multi-turn and the composite.	ReST on the original parent, with a retention threshold and revised refusal mixture.	Expected preservation fails: $G^* = .2621$, ACE-special $= .0400$, multi-turn $= .1550$.
Nemo R9, `afcbae7e`	Own `e6ad3a90` has KV $= .0516$ after second-recall training.	Return to linear memory, revise multi-turn allocation, retain ReST and the .85 threshold.	KV rises to .1290, while G^* : .4212 $\rightarrow$ .4127 and multi-turn .305 $\rightarrow$ .265.

Table 13: Same-checkpoint vectors for Nemo's referenced anchor and round-12 repair. Differences are percentage points. A shared six-coordinate cell does not mean the underlying recipes are identical.

Measurement	R8 `5ea7197d`	R12 `814a35f5`	Difference (pp)
Irrelevance	.5208	.7375	+21.67
BFCL R^*	.4260	.4394	+1.34
ACE normal	.7720	.7640	−0.80
ACE special	.5333	.3867	−14.66
GAIA text	.1262	.1359	+0.97
Composite G^*	.4986	.4823	−1.63

E.3 Recipe reuse, local repair, and intervention fidelity

Nemo's round-12 `814a35f56004` references round-8 `5ea7197d3d46`, predicts increased irrelevance performance, and changes three typed controls: `end_irr_ratio` .1 $\rightarrow$ .2, `irr_source` ToolACE to factory, and `agentic_prompt_frac` .4 $\rightarrow$ .5. Both start from base, so this is recipe reuse rather than weight inheritance. Table 13 shows the local target response and the collateral ACE-special loss. GAIA changes by one of 103 items. The two checkpoints are mutually nondominated on the four aggregate capability components, but the repair is dominated by a candidate from another arm; it is not a new global Pareto optimum. Isolating the three changes requires controlled follow-up.

The companion ReST candidate `612155ef61d3` declares `multiturn_frac=.35`, `n_mt=1500`, and `mt_source=statemachine`, expecting more multi-turn training. However, the inspected `rest_stage` consumes generated memory, search, and existing policy prompts, without adding the multi-turn pool (`ma_loop.py:905--923` at the documented revision). The shared card includes ReST in its `n_mt` activation note, exposing a description/execution mismatch. This does not make every mixture field inert: pool-size derivation and other routes can still consume them. Its $R^* = .3906$ and $G^* = .4121$ do not establish the mismatch as their cause. A corrected route would be a new experimental condition; a consumed-data manifest would make such mismatches easier to detect.

A different failure is visible in Random's ReST candidate `af26782b`: irrelevance is 1 and other BFCL components are zero, yielding $R^* = .1$ and $G^* = .05$. Its three iterations retain 430/1,037/12 records. This collapse requires examining the capability vector and retained data; checker acceptance alone does not establish reward hacking.

F COMPLETE MAIN-COMPARISON CANDIDATE REGISTER

The register preserves all 150 configurations: 147 scheduled candidates and three additional Random infrastructure records. The 141 complete official vectors each come from one checkpoint; dashes indicate missing results, not zeros. Stars mark all 72 arm–round winners by recorded G^* (ties follow ledger order). Eight-character IDs are unique in this roster; original 12-character IDs and the separate winner selection accompany the source package in `data/appendix_20260925/`.

S/C: scratch/inherited initialization; PG: grouped policy gradient; S+PG: SFT then PG. Failure codes: T, training; E, evaluation timeout; G, generation; W, teacher wait; A, infrastructure abort. The three extra Random records carry †. Infrastructure failure does not imply training ran; duplicate official sidecars are counted once.

Table 14: Complete candidate vectors and status. Stars identify all 72 round winners within the 150-record register; ACE and GAIA are search-time measurements.

Arm	Rd	ID	Route	Init	Status	R^*	G^*	ACE-N	ACE-S	GAIA
GLM	1	`961cfa82`	S+PG	S	ok	0.3935	0.4555	0.7880	0.3533	0.0874
GLM	1	`5088c115`*	S+PG	S	ok	0.4060	0.4807	0.7760	0.4933	0.0971
GLM	2	`39d1e08d`	S+PG	S	ok	0.4052	0.4834	0.7880	0.5067	0.0777
GLM	2	`0a02d4fc`*	S+PG	S	ok	0.4694	0.5152	0.8080	0.4133	0.1650
GLM	3	`8bc58b23`*	DPO	C	ok	0.2435	0.3345	0.5040	0.5333	0.0680
GLM	3	`3d487562`	S+PG	S	T	–	–	–	–	–
GLM	3	`18e64b3b`	ReST	C	ok	0.3163	0.2621	0.3920	0.0400	0.0000
GLM	4	`846e37fd`*	S+PG	S	ok	0.4611	0.5111	0.7840	0.4600	0.1553
GLM	4	`b2306dbc`	S+PG	S	ok	0.3384	0.4458	0.8080	0.4067	0.1359
GLM	5	`fe52a7ce`*	IPO	C	ok	0.4546	0.5040	0.7960	0.4467	0.1068
GLM	5	`bda2b92c`	IPO	C	ok	0.4505	0.5019	0.8040	0.4267	0.1165
GLM	6	`902c81d0`*	S+PG	S	ok	0.4445	0.5038	0.7760	0.4933	0.1359
GLM	6	`6fe40063`	S+PG	S	ok	0.4304	0.4898	0.7760	0.4400	0.1456
GLM	7	`6dc1ba17`*	S+PG	S	ok	0.4550	0.5110	0.7800	0.4733	0.1748
GLM	7	`84c56533`	S+PG	S	ok	0.3776	0.4712	0.7840	0.4467	0.1942
GLM	8	`27cb87de`*	PG	C	ok	0.4635	0.5104	0.8000	0.4400	0.1262
GLM	8	`800174c8`	S+PG	S	ok	0.4263	0.4928	0.7920	0.4733	0.1068
GLM	9	`403ac1b8`	IPO	C	ok	0.4200	0.4738	0.7760	0.4267	0.0583
GLM	9	`187fc4a6`*	S+PG	S	ok	0.4296	0.4875	0.7720	0.4600	0.1068
GLM	10	`d4ae9262`*	S+PG	S	ok	0.4426	0.5000	0.7760	0.4933	0.1068
GLM	10	`7c61cad8`	S+PG	S	ok	0.4341	0.4987	0.7960	0.4800	0.1068
GLM	11	`050bc5f2`*	S+PG	S	ok	0.4304	0.4859	0.7720	0.4533	0.0971
GLM	11	`34e86509`	S+PG	S	ok	0.4014	0.4814	0.7880	0.4800	0.1165
GLM	12	`d092b1f8`	PG	C	ok	0.4417	0.4945	0.8000	0.4200	0.1068
GLM	12	`e15712bd`*	PG	C	ok	0.4455	0.5013	0.7720	0.4733	0.1456
Qwen	1	`77013bf0`*	SFT	S	ok	0.4005	0.4540	0.7040	0.4600	0.0874
Qwen	1	`9215d88f`	SFT	S	ok	0.3748	0.4402	0.7080	0.4600	0.0680
Qwen	2	`87ba3409`	PG	C	ok	0.4097	0.4715	0.7480	0.4600	0.1068
Qwen	2	`a3ec6de9`*	ReST	C	ok	0.4231	0.4801	0.7680	0.4200	0.1359
Qwen	3	`b61baaa2`	PG	C	ok	0.4240	0.4797	0.7680	0.4333	0.1068
Qwen	3	`524566d9`*	SFT	C	ok	0.4227	0.4801	0.7760	0.4333	0.0971
Qwen	4	`0ebb8ab8`*	ReST	C	ok	0.4509	0.5002	0.7840	0.4800	0.0680
Qwen	4	`a590ecae`	ReST	C	ok	0.4444	0.4950	0.7760	0.4733	0.0777
Qwen	5	`9e4dc09e`	PG	C	ok	0.4384	0.4950	0.7880	0.4933	0.0485
Qwen	5	`a6d55136`*	SFT	C	ok	0.4502	0.5019	0.7880	0.4867	0.0680
Qwen	6	`a0722c31`	DPO	C	ok	0.4448	0.5021	0.7840	0.4867	0.1068
Qwen	6	`995a4021`*	ReST	C	ok	0.4451	0.5092	0.7800	0.5333	0.1165
Qwen	7	`6c542be2`	SFT	C	ok	0.4466	0.4942	0.7520	0.5200	0.0485

Table 14 continued; ⋆ marks a round winner.

Arm	Rd	ID	Route	Init	Status	R^*	G^*	ACE-N	ACE-S	GAIA
Qwen	7	776b0464⋆	SFT	C	ok	0.4457	0.5037	0.7800	0.5267	0.0680
Qwen	8	63f5e04c⋆	IPO	C	ok	0.4514	0.5084	0.7760	0.5200	0.1068
Qwen	8	93b15552	IPO	C	ok	0.4414	0.5035	0.7840	0.5267	0.0777
Qwen	9	5ebdd210⋆	ReST	C	ok	0.4236	0.4766	0.7400	0.4867	0.0680
Qwen	9	7b90990e	ReST	C	ok	0.4022	0.4230	0.7400	0.2333	0.0194
Qwen	10	e91e5474	SFT	C	ok	0.4475	0.4955	0.7400	0.5200	0.0874
Qwen	10	ba7cdbb6⋆	PG	C	ok	0.4431	0.4994	0.7800	0.5200	0.0485
Qwen	11	2205fb9d	DPO	C	ok	0.4396	0.4967	0.7760	0.5200	0.0485
Qwen	11	2d87781f⋆	S+PG	C	ok	0.4453	0.5034	0.7840	0.5200	0.0680
Qwen	12	27ce7ead⋆	IPO	C	ok	0.4408	0.5022	0.7800	0.5333	0.0680
Qwen	12	e7c6d715	SFT	C	ok	0.4411	0.4933	0.7400	0.5267	0.0874
Kimi	1	664544ea	DPO	S	ok	0.3733	0.3994	0.6520	0.2800	0.0777
Kimi	1	00958ed3⋆	SFT	S	ok	0.3818	0.4077	0.7360	0.1667	0.0777
Kimi	2	caf9de7b	PG	S	ok	0.3827	0.4060	0.6520	0.2667	0.1165
Kimi	2	947cc370⋆	PG	S	ok	0.4047	0.4179	0.6400	0.2667	0.1553
Kimi	3	e33d583a	ReST	S	ok	0.0621	0.0440	0.0520	0.0000	0.0000
Kimi	3	8be0bf4c⋆	PG	S	ok	0.3977	0.4415	0.6440	0.4733	0.1068
Kimi	4	aee2a138⋆	PG	S	ok	0.4022	0.4208	0.6480	0.3133	0.1068
Kimi	4	7a833133	S+PG	S	ok	0.3953	0.3625	0.5720	0.1067	0.0583
Kimi	5	a2554599⋆	PG	S	ok	0.4001	0.4258	0.6560	0.3467	0.0971
Kimi	5	a56b9076	SFT	C	ok	0.3522	0.3908	0.6480	0.2800	0.1068
Kimi	6	d3133d38	IPO	S	ok	0.3802	0.4078	0.6520	0.3000	0.0971
Kimi	6	b8475a46⋆	PG	S	ok	0.3910	0.4341	0.6440	0.4200	0.1456
Kimi	7	41e69236	PG	S	ok	0.4115	0.4334	0.6560	0.3467	0.1165
Kimi	7	b423a757⋆	PG	S	ok	0.4113	0.4353	0.6480	0.3800	0.1068
Kimi	7	084c3c78	PG	S	ok	0.3832	0.4123	0.6520	0.3133	0.1068
Kimi	8	1318c27e⋆	PG	S	ok	0.4277	0.4594	0.6560	0.4533	0.1359
Kimi	8	ebac9bfa	PG	S	ok	0.3955	0.4432	0.6480	0.4267	0.1942
Kimi	9	80c06bff	PG	S	ok	0.4110	0.4351	0.6440	0.3800	0.1165
Kimi	9	56edf0fe⋆	PG	S	ok	0.3977	0.4385	0.6440	0.4467	0.1165
Kimi	10	14cd9794⋆	PG	S	ok	0.4170	0.4491	0.6440	0.4333	0.1456
Kimi	10	9ceac02c	PG	S	ok	0.4073	0.4360	0.6520	0.3267	0.2039
Kimi	11	7feb75b3⋆	PG	S	ok	0.4094	0.4442	0.6520	0.4067	0.1553
Kimi	11	a7254599	PG	S	ok	0.3974	0.4431	0.6560	0.4133	0.1845
Kimi	12	252328bb⋆	PG	S	ok	0.4021	0.4357	0.6520	0.3933	0.1262
Kimi	12	4b255341	PG	S	ok	0.4205	0.4268	0.6520	0.2533	0.1553
Nemo	1	187ab27c⋆	PG	S	ok	0.3982	0.4208	0.6520	0.3200	0.1068
Nemo	1	175f4eb3	PG	S	ok	0.3907	0.4101	0.6480	0.3000	0.0777
Nemo	2	30240285⋆	ReST	C	ok	0.4120	0.4356	0.6560	0.3600	0.1165
Nemo	2	df5083e0	S+PG	S	ok	0.4025	0.4130	0.6240	0.3133	0.0874
Nemo	3	a0c72757	DPO	S	ok	0.3831	0.4053	0.6520	0.2800	0.0874
Nemo	3	48bafb70⋆	IPO	C	ok	0.3883	0.4099	0.6520	0.2933	0.0874
Nemo	4	bd3d5098	S+PG	S	ok	0.3956	0.3916	0.5840	0.2800	0.0583
Nemo	4	d4c9ffa7⋆	ReST	S	ok	0.3918	0.4116	0.6520	0.2867	0.0971
Nemo	5	53617f4a	ReST	C	ok	0.3895	0.4153	0.6520	0.2933	0.1359
Nemo	5	935dd59a⋆	ReST	S	ok	0.3889	0.4170	0.6560	0.2867	0.1553
Nemo	6	8fe4313f	PG	S	ok	0.3888	0.4150	0.6480	0.3133	0.1165
Nemo	6	8fbb10a3⋆	S+PG	S	ok	0.4278	0.4206	0.6120	0.3000	0.0874
Nemo	7	9ac74076	PG	S	ok	0.3851	0.4034	0.6520	0.2933	0.0388
Nemo	7	4a6f543e⋆	ReST	S	ok	0.3970	0.4074	0.6480	0.2867	0.0388
Nemo	8	edd5dd21	ReST	S	ok	0.3923	0.4070	0.6520	0.2867	0.0485
Nemo	8	5ea7197d⋆	S+PG	S	ok	0.4260	0.4986	0.7720	0.5333	0.1262
Nemo	9	6d246791	PG	C	T	–	–	–	–	–
Nemo	9	e6ad3a90⋆	ReST	S	ok	0.3992	0.4212	0.6520	0.3067	0.1262
Nemo	9	afcbae7e	ReST	S	ok	0.3959	0.4127	0.6440	0.3000	0.0874
Nemo	10	e82dd601	ReST	S	ok	0.3959	0.4147	0.6520	0.2933	0.0971
Nemo	10	cfc46f24⋆	ReST	S	ok	0.3917	0.4155	0.6560	0.2933	0.1165
Nemo	11	3806beb4	ReST	S	ok	0.3983	0.4169	0.6520	0.3000	0.0971
Nemo	11	d4e61fe3⋆	S+PG	S	ok	0.4192	0.4565	0.7120	0.4267	0.0485

Table 14 continued; ⋆ marks a round winner.

Arm	Rd	ID	Route	Init	Status	R^*	G^*	ACE-N	ACE-S	GAIA
Nemo	12	`612155ef`	ReST	S	ok	0.3906	0.4121	0.6640	0.3000	0.0583
Nemo	12	`814a35f5`⋆	S+PG	S	ok	0.4394	0.4823	0.7640	0.3867	0.1359
TPE	1	`9252a1f7`	S+PG	S	ok	0.3705	0.3372	0.4000	0.3267	0.0291
TPE	1	`253744de`⋆	IPO	S	ok	0.3808	0.4032	0.6520	0.2867	0.0680
TPE	2	`5056a7c0`	PG	S	ok	0.3626	0.3649	0.4760	0.3533	0.1165
TPE	2	`8cebf81d`⋆	DPO	S	ok	0.3850	0.4092	0.6480	0.3000	0.0971
TPE	3	`13cc2a56`⋆	IPO	S	ok	0.3867	0.4071	0.6520	0.2933	0.0680
TPE	3	`72032fcd`	S+PG	S	E	–	–	–	–	–
TPE	4	`b3b02a3d`	ReST	S	ok	0.1158	0.0619	0.0160	0.0000	0.0000
TPE	4	`98b321d8`⋆	ReST	S	ok	0.4035	0.4225	0.6560	0.3267	0.0777
TPE	5	`2462b13b`⋆	S+PG	S	ok	0.3631	0.4173	0.7720	0.2400	0.0680
TPE	5	`6436241b`	SFT	S	ok	0.4061	0.3428	0.4680	0.0933	0.0874
TPE	6	`c1c1b3d5`⋆	SFT	S	ok	0.3725	0.4150	0.6400	0.4067	0.0777
TPE	6	`d81099f6`	IPO	S	G	–	–	–	–	–
TPE	7	`9b5de393`⋆	ReST	S	ok	0.3879	0.4166	0.6680	0.3000	0.1068
TPE	7	`4725686a`	ReST	S	ok	0.3824	0.4090	0.6560	0.3067	0.0777
TPE	8	`5e7cada1`⋆	SFT	S	ok	0.4212	0.4633	0.7160	0.4267	0.0971
TPE	8	`cb2cb001`	S+PG	S	ok	0.3082	0.4197	0.7960	0.3467	0.1456
TPE	9	`c1ea089e`⋆	ReST	S	ok	0.3734	0.4104	0.6680	0.3067	0.1068
TPE	9	`eae640cc`	IPO	S	ok	0.3822	0.4068	0.6480	0.2867	0.1068
TPE	10	`b9301baf`⋆	SFT	S	ok	0.4216	0.4785	0.7600	0.4533	0.0971
TPE	10	`c748abe3`	SFT	S	W	–	–	–	–	–
TPE	11	`50801d4b`	SFT	S	ok	0.3861	0.4528	0.7520	0.4133	0.0971
TPE	11	`9e86a127`⋆	SFT	S	ok	0.4096	0.4797	0.7880	0.4933	0.0388
TPE	12	`b26c3d58`⋆	SFT	S	ok	0.3642	0.4419	0.7600	0.4267	0.0583
TPE	12	`edff2b45`	SFT	S	ok	0.3845	0.4249	0.7760	0.1800	0.1165
Random	1	`13b2362f`⋆	SFT	S	ok	0.3816	0.4267	0.6440	0.4667	0.0485
Random	1	`fd06fe1b`	S+PG	S	ok	0.3803	0.3899	0.5600	0.3333	0.0971
Random	2	`99af5226`⋆	DPO	S	ok	0.3862	0.4088	0.6520	0.2867	0.0971
Random	2	`f97ae3ff`	SFT	S	ok	0.3790	0.3824	0.5880	0.2933	0.0194
Random	3	`2d9272e9`⋆	PG	S	ok	0.3883	0.3998	0.6360	0.2333	0.1165
Random	3	`40d3c5e4`	SFT	S	ok	0.3695	0.3982	0.5960	0.3067	0.1845
Random	4	`af26782b`	ReST	S	ok	0.1000	0.0500	0.0000	0.0000	0.0000
Random	4	`dc9560c5`⋆	SFT	S	ok	0.3838	0.4567	0.7120	0.5333	0.0680
Random	5	`6f9ad72f`⋆	DPO	S	ok	0.3817	0.4046	0.6480	0.2800	0.0971
Random	5	`04ab4dc3`	S+PG	S	ok	0.3540	0.3698	0.6080	0.2267	0.0680
Random	6	`efd9bd80`	SFT	S	ok	0.4011	0.3573	0.4720	0.1933	0.0971
Random	6	`66b65582`⋆	ReST	S	ok	0.3911	0.4114	0.6520	0.3133	0.0583
Random	7	`55c1da75`	PG	S	ok	0.3855	0.4095	0.6600	0.2933	0.0777
Random	7	`0bc4e83c`⋆	S+PG	S	ok	0.4006	0.4290	0.7640	0.1800	0.1068
Random	8	`0c7150a6`⋆	PG	S	ok	0.3873	0.4064	0.6480	0.2800	0.0874
Random	8	`b8fbad7a`	DPO	S	ok	0.3834	0.4064	0.6480	0.2867	0.0971
Random	9	`c40cecd6`⋆	DPO	S	ok	0.3920	0.4176	0.6600	0.3000	0.1165
Random	9	`18460d86`	SFT	S	ok	0.3723	0.3850	0.6040	0.2867	0.0485
Random	10	`ef45e34f`⋆	SFT	S	ok	0.3952	0.3814	0.5800	0.2200	0.0583
Random	10	`8f7858fe`	S+PG	S	E	–	–	–	–	–
Random	11	`da941eb8`	IPO	S	ok	0.3757	0.3987	0.6520	0.2800	0.0583
Random	11	`bc3d4c32`⋆	IPO	S	ok	0.3861	0.4068	0.6520	0.2867	0.0777
Random	12	`5239e194`†	DPO	S	T	–	–	–	–	–
Random	12	`a802d521`	IPO	S	ok	0.3813	0.4054	0.6520	0.2933	0.0777
Random	12	`f2f55e9c`⋆	ReST	S	ok	0.3907	0.4160	0.6520	0.3000	0.1262
Random	12	`fb5a9ba7`†	PG	S	A	–	–	–	–	–
Random	12	`035ee5ac`†	SFT	S	A	–	–	–	–	–