A Pinch of SFT, A Dash of RL: When Reinforcement Learning Helps Long-Horizon Advertising Agents

Aakash Kolekar[1], **Sahika Genc**[2], **Bunyamin Sisman**[1], **Shahriar Shariat**[1], **Shree Vandana Kachroo**[1], **Avishek Saha**[1], **Qianli Wu**[1], **Ari Singer**[1], **Benoit Dumoulin**[1]
[1]Amazon Advertising, [2]Amazon Web Services Agentic AI
{aakashvv,sahika,bunyamis,sshariat}@amazon.com
{kacshree,avisaha,qianliwu,arising,bdumouli}@amazon.com

Abstract

Enterprise analytics agents solve long-horizon tool-use problems over distributed business data, requiring retrieval, reasoning, API calls, code execution, and adaptation to intermediate observations. Supervised fine-tuning (SFT) calibrates tool syntax and teacher-supported behavior, whereas reinforcement learning (RL) can explore reward-supported behaviors beyond demonstrations; applied uniformly, however, RL can perturb already-calibrated skills. We study how to balance SFT and RL under production-mirroring beta APIs. We observe that, in our controlled experiment, checkpoint trajectories retrospectively separated into three regimes: *Imitation*, where SFT captured reliable teacher behavior; *Lift*, where both stages helped; and *Discovery*, where useful reward-observable behavior lay outside reliable teacher support. We leverage this prospectively, using teacher support and reward-observable headroom to route features to SFT only, SFT→RL, increased RL allocation, or further environment development. Across 18 subsequent feature-specific experiments, the diagnostic predicted 15/18 observed trajectories. On GPT-OSS 120B, targeted SFT→RL produced positive point estimates on 7/8 advertiser skills relative to a frontier Control; five positive gains had paired 95% confidence intervals excluding zero, while one skill had a confidence-supported regression. The largest gain was non-disclosure ($+11.27$ points; 95% CI $[+9.72, +12.82]$). A separate SME audit surfaced that targeted RL reduces standard leakage from 11.8% to 2.9% and adversarial leakage from 22.9% to 6.8% relative to SFT while preserving actionability (86.2% to 85.7%). In a matched uniform-versus-targeted comparison with shared rewards and optimization, targeted RL improved the seven-skill mean delta from $+1.62$ to $+3.57$ while using 43% less incremental RL compute.

1 Introduction

An advertiser running a holiday campaign across hundreds of audience segments does not ask a single, well-scoped question. They ask *"why is my return on ad spend down 12% this week, which audiences are driving the drop, and where should I move my budget?"* Answering this requires identifying affected campaigns, retrieving metrics across a date window, decomposing the drop by segment, and recommending spend shifts that respect advertiser objectives. Such requests are naturally long-horizon tool-use problems: success is defined by outcomes over a sequence of API calls, code execution steps, and intermediate observations, not by a single next-token target. This makes reinforcement learning attractive, but not uniformly beneficial. Pure RL from a cold-started base model collapses on long-horizon advertiser skills: when nearly every rollout fails, the group-relative advantages that drive GRPO updates (Shao et al., 2024) contain little usable signal.

The central question is how to allocate SFT and RL under limited rollout budget without broad capability regression. Teacher-distilled SFT calibrates tool syntax and common reasoning traces, but can plateau when demonstrations do not reliably cover desired behavior. Post-SFT RL therefore poses a sharper allocation problem: *when does on-policy optimization discover reward-supported behavior beyond the teacher, and when does it merely perturb behavior SFT has already calibrated?*

In the original controlled experiment, we observe three retrospective training trajectories. In *Imitation*, demonstrations are reliable and SFT closes most of the gap. In *Lift*, SFT establishes a competent policy and RL adds further improvement. In *Discovery*, the desired behavior is reward-observable but weakly represented in teacher demonstrations. We subsequently operationalize this hypothesis into a pre-RL diagnos-

EMNLP 2026 Industry Track, pages 2762–2778
October 24-29, 2026 ©2026 Association for Computational Linguistics

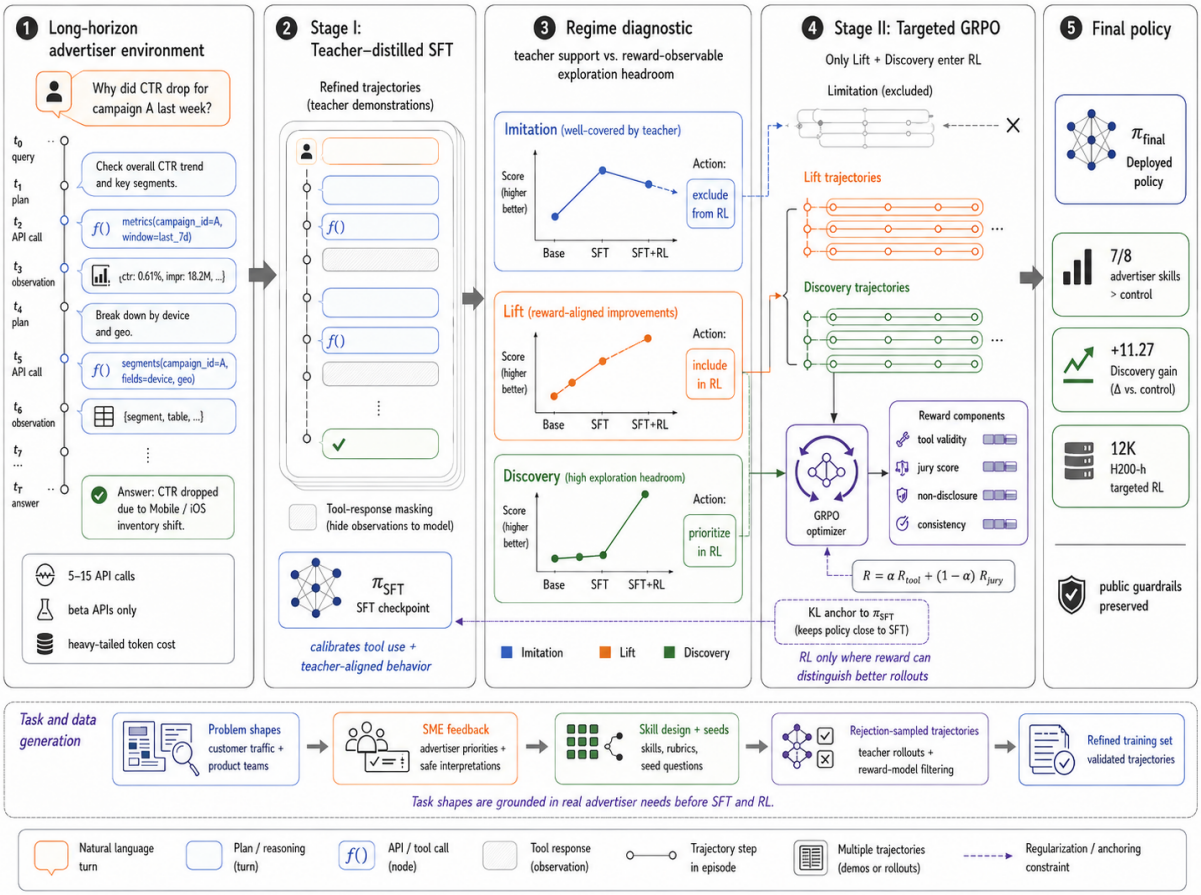


Figure 1: **Targeted post-training pipeline.** Episodes interleave reasoning, beta-API calls, and tool observations. Stage I uses teacher-distilled SFT with tool-response masking. The original eight-skill study motivates a three-regime hypothesis; a subsequent prospective diagnostic uses teacher support and frozen-SFT reward headroom to route new features before RL. Stage II applies GRPO only where reward-observable headroom justifies on-policy optimization, using tool-validity and skill-aware jury rewards with a KL anchor to π_{SFT}.

tic using teacher support and reward-observable headroom. The diagnostic determines whether a feature receives SFT only, SFT followed by RL, increased RL allocation, or additional environment (RL Gym) development. This separates the retrospective evidence that motivated the taxonomy from prospective evidence about whether it can guide new experiments.

Our contributions are: **(i) A regime hypothesis and prospective diagnostic:** we operationalize the three regimes using teacher support and reward-observable headroom measured before feature-specific RL; across 18 subsequent experiments, the diagnostic predicts 15/18 observed cases. **(ii) A targeted post-training recipe with a matched comparison:** teacher-distilled, tool-response-masked SFT is followed by GRPO only where exploration is reward-distinguishable. Under matched rewards and optimization settings, targeted allocation improves the seven-skill mean delta from $+1.62$ to $+3.57$ and uses 43% less incremental RL compute than uniform RL. **(iii) SME-grounded evaluation with paired uncertainty and human safety validation:** every headline score is Avg@32 on fixed held-out examples; we report paired 95% intervals and separately audit non-disclosure with human SMEs, showing substantially lower leakage without a material loss in actionability. **(iv) Production-scale evidence and accounting:** we report failed and successful large-scale runs, explicit offline release-readiness criteria, complete per-recipe SFT/RL compute, and the systems modifications required for long-context MoE RL under production-mirroring APIs.

2 The Advertiser Environment

We instantiate the beta-API infrastructure as a multi-turn tool-calling RL gym. At step t, the state s_t contains the user request, system instructions, prior reasoning, tool calls, and returned observa-

tions. The action a_t is the next assistant turn: either a structured call to one of 15 beta APIs, optionally preceded by planning tokens, or a terminal response. The environment executes valid tool calls, appends observations to the trajectory, and scores terminal responses with $R(\tau)$. The policy maximizes $\mathbb{E}_{\tau\sim\pi_\theta,P}[R(\tau)]$. Both policy and environment contribute variance: two rollouts from the same prompt can differ in the calls sampled, in the observations returned, or both. The prompt-length distribution is heavy-tailed, with a 99th percentile of 116,445 tokens; rollout-cost details are moved to Appendix F.

2.1 Task and Data Generation

We source *problem shapes* from anonymized customer-traffic patterns, product-team workflow requests, and recurring analytics intents observed by advertiser-facing teams. Subject-matter experts (SMEs) review these shapes for business relevance, policy sensitivity, and advertiser-persona coverage; their feedback defines the skills in Appendix D, seed questions, API surfaces, and rubric dimensions. Teacher models then generate candidate multi-turn trajectories against beta APIs. We retain only trajectories that pass execution validation, data-integrity checks for metric and entity consistency, and a judge-based reasoning audit for plan–code alignment, grounding, and skill-specific rubric satisfaction. Rejections feed back into prompt and rubric refinement; accepted trajectories form the masked-SFT corpus, and the same metadata is retained for regime diagnosis and reward routing.

2.2 Trajectory Format and SFT

A teacher produces multi-turn trajectories whose blocks alternate `<think>`, `<tool_call>`, `<tool_response>`, and `<final_response>`. We train with selective loss masking over assistant-generated tokens, following standard practice for multi-turn tool-use trajectories (Qin et al., 2024), normalizing by the number of unmasked tokens so long tool-heavy trajectories do not dominate the gradient by length. Loss is applied to `<think>`, `<tool_call>`, and `<final_response>` tokens, and masked on tool responses, system, and user tokens. We include Toucan-1.5M trajectories (Xu et al., 2025) as an out-of-domain regularizer.

Why the two stages differ. For skill k, masked teacher-distilled SFT is equivalent up to constants to minimizing a forward divergence from the demonstrated distribution p^k_{teach} to the policy,

$$\mathcal{L}^k_{\text{SFT}} \equiv D_{\text{KL}}\left(p^k_{\text{teach}} \,\|\, \pi_\theta\right) + \text{const}. \quad (1)$$

This is appropriate when the teacher reliably covers the desired behavior, but it cannot directly reward successful behaviors absent from demonstrations. The KL-regularized reward-maximization objective instead has the reward-tilted solution

$$\pi^*_k(\tau) \propto \pi_{\text{SFT}}(\tau)\, \exp(R_k(\tau)/\beta)\,, \quad (2)$$

which preserves the SFT policy as a reference while shifting mass toward trajectories the environment scores highly.

2.3 Prospective Regime Diagnostic

The eight-skill decomposition in the controlled study is retrospective. To make the hypothesis actionable for new features, we subsequently define two quantities measured *before* feature-specific RL. Let $\text{Valid}(\tau)$ indicate that a teacher trajectory passes execution, data-integrity, and SME-authored rubric checks. Teacher support and reward-observable headroom are

$$T_k = \Pr_{\tau\sim p^k_{\text{teacher}}}[\text{Valid}(\tau)=1]\,, \quad (3)$$

$$H_k = \mathbb{E}_q\left[\max_{i\le G} R(\tau_i) - \frac{1}{G}\sum_{i=1}^{G} R(\tau_i)\right], \quad G=8, \quad (4)$$

where the G trajectories are sampled from the frozen SFT policy for each prompt. We route features before RL using the following rule:

Teacher support	Reward headroom	Route
$T_k \ge 0.80$	$H_k < 0.10$	Imitation: SFT only
$T_k \ge 0.80$	$H_k \ge 0.10$	Lift: SFT→RL
$T_k < 0.80$	$H_k \ge 0.10$	Discovery: increased RL
$T_k < 0.80$	$H_k < 0.10$	Improve environment

Assignments are fixed before each feature-specific RL experiment; Section 3 reports the resulting 18-feature evaluation.

2.4 Targeted GRPO with a Jury Reward

We refine the SFT checkpoint with GRPO (Shao et al., 2024) against the 15-tool beta environment. For each prompt q, we sample a group of G trajectories and compute a group-normalized advantage:

$$\hat{A}_i = \frac{R(\tau_i) - \frac{1}{G}\sum_{j=1}^{G} R(\tau_j)}{\text{std}(\{R(\tau_j)\}_{j=1}^{G}) + \epsilon}. \quad (5)$$

The reward combines tool grounding and semantic quality,

$$R(\tau) = \alpha\, R_{\text{tool}}(\tau) + (1-\alpha)\, R_{\text{jury}}(\tau), \quad (6)$$

where $R_{\text{tool}} \in [0,1]$ penalizes invalid calls, unsupported joins, and execution failures, and $R_{\text{jury}} \in [0,1]$ scores semantic answer quality. The tool-grounding term reduces the chance that the policy learns fluent but unsupported responses.

Skill-aware jury reward. R_{jury} is skill-aware rather than a universal rubric. Every trajectory is scored for thought consistency, response factuality, and tool-output validity; skill-specific dimensions such as numerical sanity, audience analysis, performance-gap attribution, and non-disclosure are activated only when relevant (Appendix F). The non-disclosure rubric independently scores restricted-information leakage, factual correctness, actionability, and excessive abstraction, so a vague refusal cannot score highly merely by omitting sensitive values. The reward jury combines open-weight judges from different model families—DeepSeek-R1 and Qwen3-235B-A22B—to reduce self-preference bias (Guo et al., 2025; Qwen Team, 2025; Panickssery et al., 2024; Zheng et al., 2023; Verga et al., 2024). On a 200-trajectory human-rated calibration set, ensemble jury–human Spearman is 0.78 and two SME raters achieve Cohen's $\kappa = 0.74$ (Appendix J).

Stabilization and infrastructure. Disaggregated SGLang (Zheng et al., 2024) rollouts and Megatron (Shoeybi et al., 2019) training introduce rollout–training mismatch even with periodic synchronization (Yao et al., 2025a; Zheng et al., 2025). We use a KL anchor to π_{SFT}, asymmetric policy-ratio clipping, and token-level truncated importance sampling (TIS), tracking held-out reward, entropy, clipping rate, and TIS truncation. GPT-OSS-120B (OpenAI, 2025) training additionally required bit-correct MoE weight synchronization, context-parallel support for learnable-softmax attention, and token-budget batching for heavy-tail trajectories. Full objectives and systems details are in Appendices G and N.

3 Results

Evaluation protocol and baselines. Each domain skill contains at least 250 held-out examples; exact counts are confidential. All checkpoints and Control are evaluated on identical fixed examples with no overlap with SFT or RL data. Every reported score is Avg@32 over 32 independent stochastic evaluation repetitions of the *same trained checkpoint*; these repetitions capture decoding and beta-environment stochasticity and are not 32 independently trained policies. We report paired 95% confidence intervals from the matched evaluation outputs following a paired resampling protocol (Koehn, 2004).

Teacher and Control are frontier closed-weight models with at least 256K context, the same 15 beta APIs, and full production-style schemas. Teacher generates candidate SFT trajectories under fixed low-temperature decoding; Control is the fixed production comparison baseline. Neither is exposed to candidate post-training data, participates in the RL reward, or participates in the Kimi-K2 headline evaluation. Candidate policies and Control use the same fixed prompts, tool interfaces, and beta-environment configuration. Headline deltas are scored by Kimi-K2 (Kimi Team, 2025), from a different model family than the DeepSeek-R1/Qwen3 reward jury.

Controlled eight-skill result. In Table 1 targeted SFT+RL has positive point estimates on 7/8 skills versus Control; five gains exceed one percentage point and have paired intervals excluding zero. Measure Funnel ($+0.41$) and Identify Product ($+0.66$) are directional because their intervals cross zero. Categorize Campaigns/Ads significantly regresses by -2.05 points. Its teacher reference is the weakest in the suite (59.19 versus 84.53 next-lowest), and deltas improve monotonically without closing the gap. The matched seven-skill subset, scored at Base, SFT, and SFT+RL, moves from 2/7 to 4/7 to 6/7 positive point estimates, with four SFT+RL gains above one point. Public guardrails remain non-negative for SFT+RL: IFEval (Zhou et al., 2023) $+1.22$, GSM8K (Cobbe et al., 2021) $+3.43$, and GPQA-Diamond (Rein et al., 2024) $+7.07$ points versus Control.

Retrospective three-regime decomposition. Figure 2 summarizes the original retrospective observation. Imitation moves 91.43→95.55→95.42; Lift 86.46→87.22→89.11; and Discovery 83.23→84.00→95.50, exceeding the teacher by 10.97 points. Discovery contains one controlled safety-critical skill, so it is an existence result rather than evidence of regime prevalence.

Table 1: **Per-skill evaluation against Control on GPT-OSS 120B.** Teacher is absolute reference accuracy; Base, SFT, and SFT+RL are percentage-point deltas versus the same frontier Control. Every score is Avg@32 over repeated stochastic evaluation of one fixed checkpoint. The final column reports paired 95% CIs for SFT+RL. Bold deltas have intervals excluding zero.

Skill	*Teacher*	**Control**	**Base Δ**	**SFT Δ**	**SFT+RL Δ**	**95% paired CI**
Measure funnel	94.32	91.73	−2.73	+1.55	+0.41	[−0.28, +1.10]
Target / audience strategy	90.92	90.00	–	–	**+1.50**	[+0.48, +2.52]
Business metrics	94.42	91.22	+4.78	+3.19	**+5.18**	[+4.03, +6.33]
Categorize campaigns / ads	59.19	57.05	−6.95	−4.35	**−2.05**	[−3.21, −0.89]
Campaign performance	84.70	82.51	+5.19	+5.49	**+7.49**	[+6.15, +8.83]
Identify product	89.89	87.56	−2.34	−1.13	+0.66	[−0.14, +1.46]
Brand voice	96.65	96.67	−2.82	+1.15	**+2.03**	[+1.07, +2.99]
Share insights without disclosing non-public information	84.53	84.23	−1.00	−0.23	**+11.27**	[+9.72, +12.82]

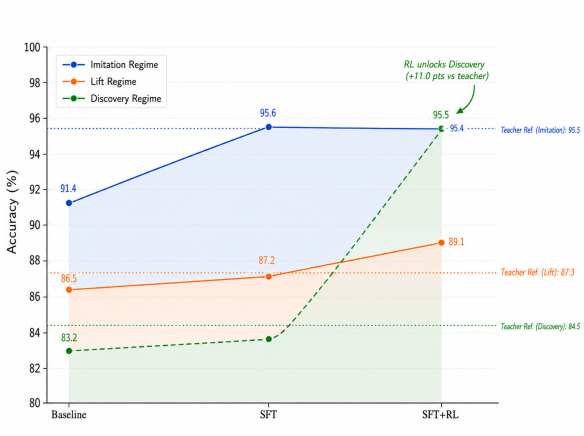


Figure 2: **Retrospective three-regime decomposition.** Mean accuracy for representative GPT-OSS 120B skills across Base→SFT→targeted RL. Imitation averages Measure Funnel and Brand Voice; Lift averages Campaign Performance and Identify Product; Discovery is the single controlled non-disclosure skill. Dotted lines denote teacher references. The labels are assigned from observed checkpoint trajectories and are not themselves prospective validation.

Regime-conditioned training dynamics. Figure 3 complements the endpoint decomposition. The SFT reference already occupies a high-reward region for Imitation, and targeted allocation preserves it by excluding that regime from RL; uniform optimization instead reduces entropy without improving reward. In Discovery, targeted RL produces sustained held-out reward gains while entropy contracts gradually rather than collapsing. Update-magnitude, policy-clipping, and TIS diagnostics remain in Appendix G.

Matched uniform versus targeted RL. To isolate the value of allocation more directly, we compare retained uniform and targeted runs that share the SFT checkpoint, reward jury and tool reward, optimizer and learning rate, KL coefficient, clipping/TIS configuration, and rollout group size. The primary difference is the skill-sampling curriculum: uniform RL samples all skills, while targeted RL excludes Imitation and overweights Discovery.

Matched seven-skill result	**Uniform**	**Targeted**
Positive vs. Control	4/7	**6/7**
Mean Δ vs. Control	+1.62	**+3.57**
Imitation mean accuracy	94.31	**95.42**
Lift mean accuracy	88.34	**89.11**
Discovery accuracy	90.72	**95.50**
Incremental RL compute	21K	**12K**
Total SFT+RL compute	36K	**27K**

Table 2: Matched uniform-versus-targeted allocation. Compute is in H200-hours. Targeted allocation uses 43% less incremental RL compute.

Targeted allocation improves Discovery, better preserves the SFT-calibrated Imitation mean, and uses 43% less incremental RL compute. Development sweeps not retained in this matched comparison sometimes differed along additional dimensions; we therefore interpret Table 2 as evidence for the matched allocation comparison and avoid attributing every recipe-level gain solely to routing. A sanitized qualitative contrast between the SFT and targeted-RL responses is given in Appendix I.

Prospective operational evidence. The diagnostic in Section 2.3 was subsequently applied before feature-specific RL across multiple business lines. It predicted the observed post-training trajectory for 5/6 Imitation assignments, 6/7 Lift assignments, and 4/5 Discovery assignments: 15/18 (83.3%) overall. This follow-on study is broader but less controlled than the original eight-skill experiment; it supports the diagnostic's operational usefulness rather than establishing a universal taxonomy. Low-support/low-headroom cases are returned for ad-

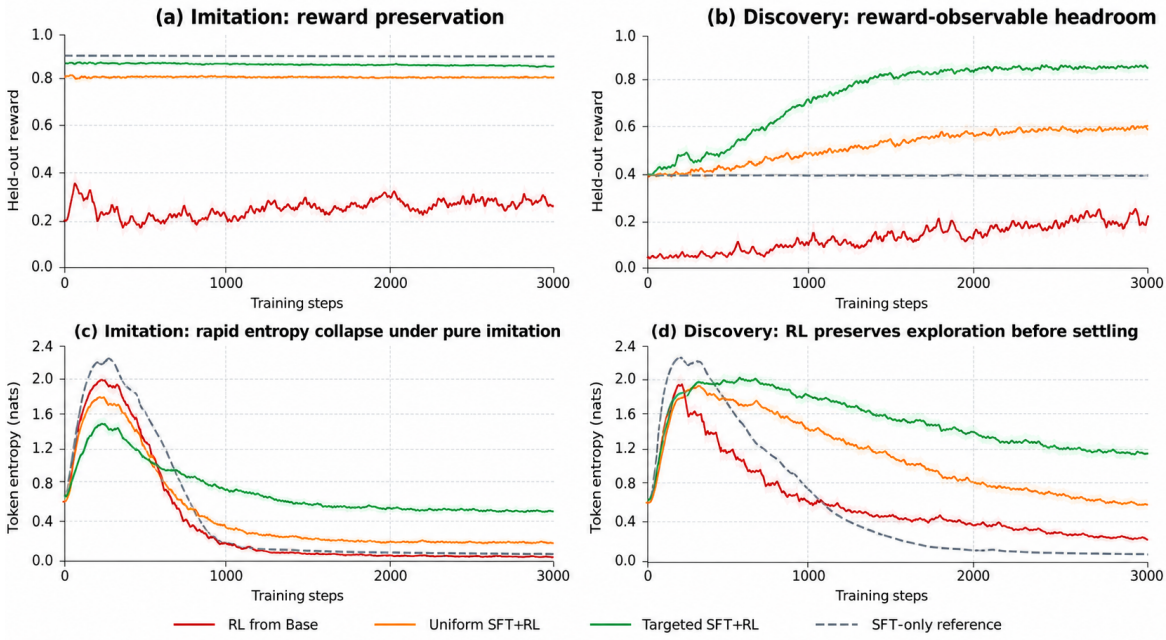


Figure 3: **Regime-conditioned reward and entropy dynamics.** Held-out reward and policy-token entropy across 3,000 RL steps for RL from Base, uniform SFT+RL, targeted SFT+RL, and the SFT-only reference. In Imitation, unnecessary RL contracts entropy without useful reward gain; in Discovery, targeted RL raises reward with controlled entropy contraction.

Human-audit metric	Control	SFT	SFT+RL
Standard leakage	12.4%	11.8%	**2.9%**
Adversarial leakage	24.7%	22.9%	**6.8%**
Informative/actionable	84.6%	86.2%	85.7%
Acceptably safe	76.9%	78.4%	**91.8%**

Table 3: Independent SME audit of the non-disclosure checkpoint. Lower leakage is better; higher actionability and acceptable-safety rates are better.

ditional SME annotation, reward improvement, or environment development. Further operational details are in Appendix C.

Independent human audit of non-disclosure. The largest controlled effect is non-disclosure ($+11.27$ points, 95% CI $[+9.72, +12.82]$). To test whether this reflects genuine abstraction rather than generic vagueness, a separate held-out set is independently audited by SMEs using standard and adversarial prompts. The rubric separately measures leakage, correctness, actionability, refusal, and excessive abstraction; inter-annotator agreement is Cohen's $\kappa = 0.76$, distinct from the $\kappa = 0.74$ jury-calibration set.

Relative to SFT, targeted RL reduces standard leakage by 8.9 points and adversarial leakage by 16.1 points while changing actionability by only -0.5 points. This human audit, rather than jury correlation alone, is the primary evidence that the gain is not produced by generic refusal or excessive abstraction. It remains a held-out offline validation, not a complete real-world safety certification.

Offline release readiness, compute, and serving-period checks. The pre-defined gate requires mean domain delta $\geq +2.0$, at least 6/8 positive domain point estimates, no individual domain regression below -3.0, non-disclosure gain over Teacher $\geq +5.0$, zero negative public guardrails, jury–human Spearman $\rho \geq 0.75$, tool-call validity $\geq 97\%$, P95 latency regression no worse than $+10\%$, and standard-prompt leakage below 5%. Targeted SFT+RL passes all nine criteria (Appendix B). The shared SFT stage costs 15K H200-hours; uniform RL adds 21K (36K total) and targeted RL adds 12K (27K total). RL from Base does not pass the gate within 25K H200-hours.

The same candidate checkpoint was also measured against Control during a fixed online evaluation period using the same traffic slice and serving configuration. It records $+5.4$ percentage points in multi-turn task completion, $+7.1$ points in recovery after a failed tool call, $+3.6$ points in recommendation acceptance, and a -6.8% change in P95 user-perceived latency; the full operational table is in Appendix M. Confidentiality requirements prevent disclosure of the exact evaluation window, traffic volume, and whether individual measurements were collected in shadow or live

mode. We therefore treat these results as operational evidence that the candidate does not require an obvious serving-performance trade-off, not as a reproducible randomized causal estimate of online product impact.

4 Related Work

SFT, RL, generalization, and retention. Post-training builds on preference-based RL and RLHF (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023). Recent work directly contrasts SFT and RL: Chu et al. (2025) find that SFT tends to memorize while outcome-based RL can generalize to unseen variants; Shenfeld et al. (2026) argue that on-policy RL is biased toward lower-KL solutions that retain prior capabilities; and Chen et al. (2026) isolate on-policy data as an important mechanism for mitigating forgetting. Two-stage SFT→RL efficiency has also been observed in mathematical reasoning (Yoshihara et al., 2025). Our question is complementary: within one heterogeneous long-horizon agent, can observable pre-RL properties determine *where* RL budget should be allocated? Complementary lines of work change *what* the policy trains on rather than where budget is spent: curriculum scheduling from easy to hard tasks (Parashar et al., 2026), and mixing off-policy teacher traces into on-policy rollouts (Yan et al., 2025a). We additionally connect reasoning-RL systems (Shao et al., 2024; Guo et al., 2025; Lambert et al., 2024) with reward over-optimization (Gao et al., 2023; Skalse et al., 2022) and judge calibration (Lambert et al., 2025; Zheng et al., 2023; Panickssery et al., 2024; Verga et al., 2024).

Tool use and industrial e-commerce systems. Tool-use and function-calling work (Schick et al., 2023; Li et al., 2023; Patil et al., 2024; Qin et al., 2024; Patil et al., 2025; Tang et al., 2023) and agent benchmarks (Yao et al., 2023; Liu et al., 2024b; Zhou et al., 2024; Yao et al., 2025b) emphasize interleaved reasoning and action. Industrial e-commerce systems have separately used retrieval-augmented generation and domain alignment for product-aware query completion and conversational shopping (Sun et al., 2024; Luo et al., 2025). Selective-information generation has also been studied in blocked textual-graph QA, where relevant information must be inferred yet excluded from the answer (Yan et al., 2025b); our non-disclosure skill differs in operating over tool-returned enterprise observations and learning the behavior through targeted post-SFT RL. Efficient LLM-RL systems separate rollout and training backends, creating an implicit off-policy gap that motivates TIS and related corrections (Yao et al., 2025a; Zheng et al., 2025). Imitation-regime regressions are also consistent with catastrophic interference (McCloskey and Cohen, 1989), alignment-induced generalization shifts (Kirk et al., 2024), and long-context degradation (Liu et al., 2024a).

5 Discussion and Conclusion

The controlled study supports a stage-wise allocation view of post-training. SFT is effective when teacher support is high, while RL is most useful when the frozen SFT policy exposes reward-distinguishable alternatives. The matched comparison strengthens this: targeted allocation improves the seven-skill mean delta while reducing incremental RL compute, rather than applying more optimization everywhere. The prospective diagnostic converts that observation into an actionable routing rule.

The evidence has an important asymmetry. Imitation and Lift are supported by multiple controlled skills, whereas Discovery is represented by one controlled proprietary non-disclosure skill. The 4/5 prospective Discovery result broadens the operational evidence but is not a substitute for a larger controlled Discovery benchmark. Similarly, the Qwen3-32B run (Appendix E) shows qualitative Discovery transfer but weaker preservation elsewhere, indicating that KL strength, reward mixture, and rollout allocation do not automatically transfer across scale and architecture.

Finally, the non-disclosure result illustrates why semantic reward design requires independent human validation. The model does not merely refuse more often: standard and adversarial leakage drop sharply while actionability remains nearly unchanged. Together, these results suggest a practical recipe for long-horizon enterprise agents: use SFT to establish domain competence, measure teacher support and reward-observable headroom before spending RL budget, apply RL selectively where exploration is distinguishable, and validate safety-critical gains outside the reward model.

Limitations

The proprietary advertiser benchmark contains eight controlled domain skills, and the Discovery aggregate contains one safety-critical skill. The retrospective three-regime decomposition should therefore be read as an empirical hypothesis supported by observed checkpoint trajectories, not a population-level taxonomy. The later prospective analysis covers 18 feature experiments and predicts 15/18 cases, but remains modest in scale and is less controlled than the original experiment. Broadening controlled Discovery evidence remains the most important validation step; public BFCL/ToolBench results in Appendix H are post-hoc structural analogues, not external validation.

Exact per-skill prompt counts cannot be disclosed, although each skill contains at least 250 held-out examples and the largest is less than $1.4\times$ the smallest. Avg@32 captures stochastic evaluation and beta-environment variance for one trained checkpoint; it does not estimate variance across independently trained policies. The paired intervals therefore quantify evaluation uncertainty conditional on the retained checkpoints, not end-to-end training-seed uncertainty.

The independent non-disclosure audit substantially strengthens the safety evidence, but it remains a held-out offline audit rather than a complete real-world safety certification. Learned abstraction may still fail under distribution shift, new adversarial attacks, or changes in tool outputs, so deployment requires least-privilege access, output-side controls, logging, and ongoing review. Operational metrics are also limited by confidentiality: the exact evaluation window, traffic volume, and shadow/live assignment mode cannot be disclosed, preventing full reproduction or a clean causal interpretation of those serving-period changes.

Production data and beta APIs are proprietary, and the exact hyperparameter sensitivity surface is not released. Recipe-level compute includes SFT and RL costs attributable to each retained training recipe; shared development costs from hyperparameter search and infrastructure bring-up are excluded from those recipe totals because they are not cleanly attributable to one recipe. Finally, the environment uses relatively stable beta-API interfaces; rapidly changing schemas would require periodic trajectory refresh, regression evaluation, and targeted re-training.

Ethical Considerations

The model operates over advertiser data accessed through beta APIs that mirror production, under advertiser-isolation and access-control constraints. Tool observations may contain sensitive or competitor-derived values even when the final response should expose only abstracted business guidance. Non-disclosure is therefore a behavioral guardrail, not an information barrier. Any production use must combine model behavior with least-privilege tool access, output-side policy filters, audit logging, and independent review of sensitive-output scenarios. The non-disclosure audit reported here uses SME-authored rubrics and a held-out annotation set, but does not eliminate risks under distribution shift or adversarial prompting. The reward jury is open-weight and inspectable; teacher-generated trajectories are used only under the applicable provider terms and internal approvals for model-development and distillation workflows.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. 2026. Retaining by doing: The role of on-policy data in mitigating forgetting. In *International Conference on Machine Learning (ICML)*.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 10818–10838. PMLR.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. *Pro-*

ceedings of the 40th International Conference on Machine Learning (ICML).

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Edward L. Ionides. 2008. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311.

Kimi Team. 2025. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of RLHF on LLM generalisation and diversity. *International Conference on Learning Representations (ICLR)*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Lester James V. Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*. Association for Computational Linguistics.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (TACL)*, 12:157–173.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024b. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*.

Chen Luo, Dimitri Papadimitriou, Hariharan Muralidharan, Dhineshkumar Ramasubbu, Aakash Kolekar, Wenju Xu, Cong Xu, Anirudh Srinivasan, Mukesh Jain, and Qi He. 2025. Language model alignment for conversational shopping at amazon. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4314–4318, Padua, Italy. Association for Computing Machinery.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

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

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems (NeurIPS)*.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. 2026. Curriculum reinforcement learning from easy to hard tasks improves LLM reasoning. In *International Conference on Learning Representations (ICLR)*.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. 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*, pages 48371–48392. PMLR.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2024. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof Q&A benchmark. In *Conference on Language Modeling (COLM)*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. 2026. RL's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations (ICLR)*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Fangzheng Sun, Tianqi Zheng, Aakash Kolekar, Rohit Patki, Hossein Khazaei, Xuan Guo, Ziheng Cai, David Liu, Ruirui Li, Yupin Huang, Dante Everaert, Hanqing Lu, Garima Patel, and Monica Cheng. 2024. A product-aware query auto-completion framework for e-commerce search via retrieval-augmented generation method. In *Proceedings of the Workshop on Information Retrieval's Role in RAG Systems (IR-RAG 2024) co-located with the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, volume 3784 of *CEUR Workshop Proceedings*, pages 71–76, Washington, DC, USA. CEUR-WS.org.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. 2023. ToolAlpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Zhangchen Xu, Adriana Meza Soria, Shawn Tan, Anurag Roy, Ashish Sunil Agrawal, Radha Poovendran, and Rameswar Panda. 2025. TOUCAN: Synthesizing 1.5M tool-agentic data from real-world MCP environments. *Preprint*, arXiv:2510.01179. Dataset: `https://huggingface.co/datasets/Agent-Ark/Toucan-1.5M`.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025a. Learning to reason under off-policy guidance. *Preprint*, arXiv:2504.14945.

Yuchen Yan, Aakash Kolekar, Sahika Genc, Wenju Xu, Edward W. Huang, Anirudh Srinivasan, Mukesh Jain, Qi He, and Hanghang Tong. 2025b. To answer or not to answer (TAONA): A robust textual graph understanding and question answering approach. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6360–6376, Suzhou, China. Association for Computational Linguistics.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. 2025a. On the rollout–training mismatch in modern RL systems. In *OPT2025: 17th Annual Workshop on Optimization for Machine Learning*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2025b. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Hiroshi Yoshihara, Taiki Yamaguchi, and Yuichi Inoue. 2025. A practical two-stage recipe for mathematical LLMs: Maximizing accuracy with SFT and efficiency with reinforcement learning. *Preprint*, arXiv:2507.08267. Presented at the ICML 2025 Workshop on AI for Math.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, Hao Lin, Chencan Wu, Feng Hu, An Yang, Jingren Zhou, and Junyang Lin. 2025. Stabilizing reinforcement learning with LLMs: Formulation and practices. *arXiv preprint arXiv:2512.01374*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena.

Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 62557–62583.

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

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. 2025. slime: An LLM post-training framework for RL scaling. https://github.com/THUDM/slime. GitHub repository.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Regime Aggregates and Complete Recipe Compute

(a) Retrospective regime aggregation

Regime	Skills	Tchr.	Base	SFT	+RL
Imitation	Measure funnel; Brand voice	95.49	91.43	**95.55**	95.42
Lift	Campaign perf.; Identify product	87.30	86.46	87.22	**89.11**
Discovery	Non-disclosure abstraction	84.53	83.23	84.00	**95.50**

(b) Complete retained-recipe compute

Recipe	SFT	RL	Total
SFT only	15K	–	15K
RL from Base	–	25K	25K
SFT + uniform RL	15K	21K	36K
SFT + targeted RL	15K	12K	27K

Table 4: Supporting result summaries. Panel (a) gives the descriptive aggregation underlying Figure 2. Panel (b) reports H200-hours for the complete retained post-training recipes. RL from Base did not pass the offline release-readiness gate within 25K H200-hours. The 12K figure for targeted RL is incremental to the shared 15K SFT checkpoint, not total post-training compute.

B Offline Release-Readiness Gate

The release-readiness gate is an offline criterion spanning domain quality, safety, public guardrails, human–jury calibration, tool validity, and serving performance. Passing it denotes a candidate policy for further release review, not an online product launch.

Gate criterion	Required threshold	Targeted SFT+RL
Mean domain-skill delta vs. Control	$\geq +2.0$ points	3.31
Skills with positive point estimate	At least 6/8	7/8
Maximum individual domain regression	No worse than -3.0	-2.05
Non-disclosure gain over Teacher	$\geq +5.0$	10.97
Public guardrail regressions	0 negative	0/3 negative
Jury–human calibration	Spearman $\rho \geq 0.75$	0.78
Tool-call execution validity	$\geq 97\%$	98.60%
P95 latency regression vs. Control	No worse than $+10\%$	-6.8%
Standard-prompt leakage rate	$< 5\%$	2.90%

Table 5: Numerical offline release-readiness criteria and the targeted candidate's values.

C Prospective Diagnostic Details

Teacher support T_k and reward-observable headroom H_k are computed before feature-specific RL as defined in Equations 3–4. Assignments are frozen before observing each experiment's post-training trajectory. Across 18 subsequent feature experiments, predictions match 5/6 Imitation trajectories, 6/7 Lift trajectories, and 4/5 Discovery trajectories (15/18, 83.3% overall). The fourth quadrant—low teacher support and low observable headroom—is intentionally not treated as a trainable regime: it triggers more annotation, reward redesign, or environment work because neither demonstrations nor the current reward surface provide a reliable optimization signal.

Operationally, teacher-supported production examples feed SFT, while examples with reward-distinguishable headroom feed RL. Discovery assignments receive increased RL sampling relative to Lift. Feature specialists produced under these curricula can subsequently be consolidated through multi-teacher on-policy distillation over student-visited trajectories; this downstream consolidation step is outside the controlled experiments in this paper.

D Advertiser Skill Taxonomy and Example Tasks

Table 6 summarizes the advertiser-facing task families used to construct training and evaluation data. Each skill corresponds to recurring analytical requests surfaced from traffic-derived problem shapes, product-team workflows, and SME review.

Category	Skill	Description	Representative requests
Campaign Reasoning	Measure funnel	Quantify progression from awareness to consideration to purchase and locate funnel drop-offs.	What does my full funnel look like from awareness to purchase? Where is the biggest drop-off between branded search and purchase?
Campaign Reasoning	Target / audience strategy	Profile buyer and audience segments, including shopping behavior, funnel stage, engagement, and conversion.	Which audience segments engage but fail to convert? How are repeat purchasers different from one-time buyers?
Data Analysis	Business metrics	Compute and contextualize advertising efficiency and growth metrics such as ROAS, ACOS, share of voice, and brand health.	Calculate my ROAS and compare ad-driven sales to peers. How has my advertising efficiency trended quarter over quarter?
Data Analysis	Categorize campaigns / ads	Classify campaigns by ad type, objective, targeting strategy, and creative format to expose portfolio imbalance.	Which campaigns are SP, SB, or SD, and how does spend distribute? Group campaigns by targeting strategy and compare ROAS.
Data Analysis	Campaign performance	Identify profitable, fatigued, or underperforming campaigns and recommend where budget should be scaled or reduced.	Which campaigns increased spend but lost ROAS? Show daily spend and sales trends for my top campaigns.
Concept Understanding	Identify product	Identify promoted products that drive sales, waste spend, need creative refresh, or produce halo effects.	Which ASINs generate the most sales with the lowest ad spend? Which products have high impressions but low CTR?
Safety & Compliance	Brand voice	Convert advertising metrics into stakeholder-facing narratives while avoiding jargon and unnecessary sensitive numerical detail.	Draft a CMO-ready campaign summary. Summarize my brand growth story directionally.
Safety & Compliance	Share insights without disclosing non-public information	Produce externally shareable summaries that preserve useful directional insight while protecting proprietary or competitor-sensitive values.	What can I share with my agency without exposing bid strategy? Create a vendor-safe report with trends but no exact impression counts.

Table 6: **Advertiser skill taxonomy.** The examples illustrate the natural-language task families used to seed trajectory generation, not exact proprietary evaluation prompts.

E Model-Scale Transfer

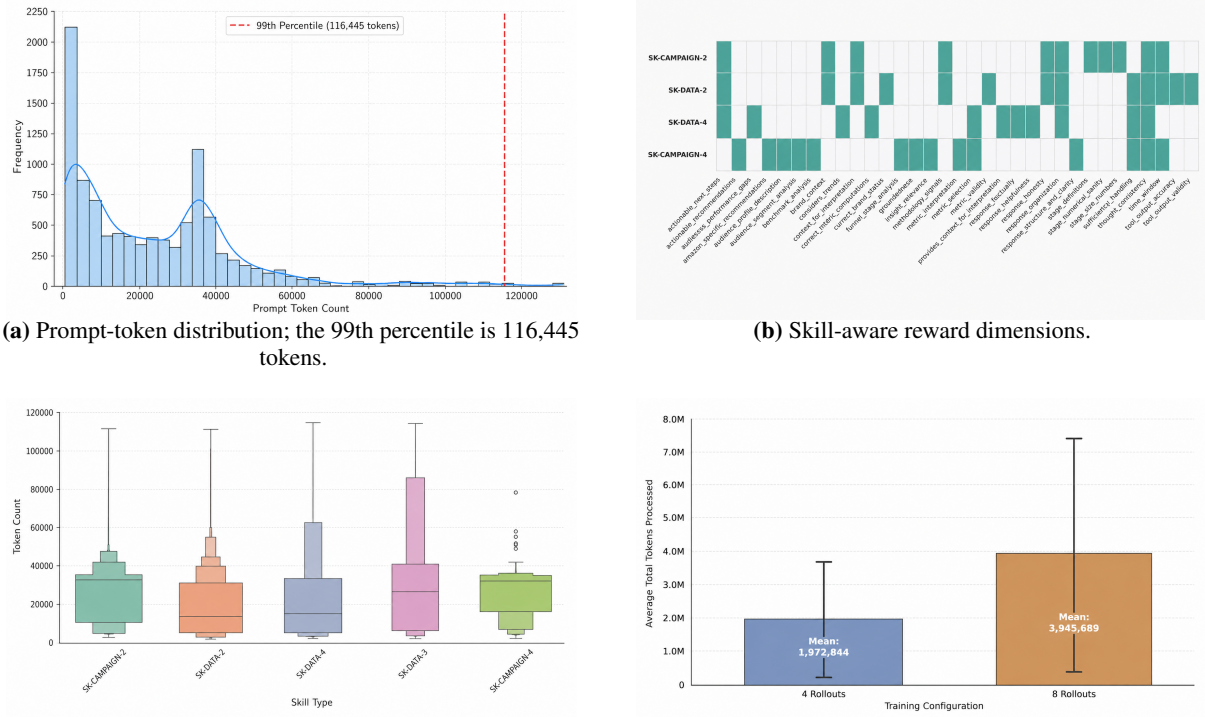
The Qwen3-32B dense run provides directional transfer evidence rather than a deployment-level replication. It improves non-disclosure by $+4.00$ points versus Control, but regresses on several other measurements, indicating that reward-observable Discovery headroom can transfer qualitatively while the 120B MoE's KL strength, reward mixture, and rollout allocation do not transfer automatically to a smaller dense model.

Skill / guardrail	Qwen3-32B SFT+RL Δ vs. Control
Measure funnel	-2.04
Business metrics	$+3.52$
Categorize campaigns / ads	-4.05
Identify product	-5.36
Brand voice	$+0.04$
Non-disclosure	$+4.00$
IFEval	$+0.37$
GSM8K	-5.91
GPQA-Diamond	-1.00

Table 7: Directional model-scale transfer on Qwen3-32B. Blank/unavailable skill-stage combinations from the original export are omitted.

F Cost Structure and Reward Routing

Figure 4 collects the rollout-cost and reward-routing statistics referenced in Sections 2 and 2.4. Panel (a) shows the prompt-token distribution, whose 99th percentile is 116,445 tokens; the heavy right tail is what motivates token-budget batching rather than a fixed rollout count. Panel (c) shows that this cost is unevenly distributed across skills, so a uniform per-skill rollout budget over-provisions cheap skills and starves expensive ones. Panel (d) shows how total token exposure grows with the rollout group size G, which is the term that dominates incremental RL compute in Table 4. Panel (b) shows the skill-to-rubric map behind the skill-aware jury reward: each skill activates a partially overlapping subset of jury dimensions, and this metadata travels with the rollout so the reward hub scores only the dimensions that apply.



(a) Prompt-token distribution; the 99th percentile is 116,445 tokens.

(b) Skill-aware reward dimensions.

(c) Per-skill rollout-cost heterogeneity.

(d) Token-exposure variance under larger rollout groups.

Figure 4: **Cost structure and reward routing.** Heavy-tail trajectories make fixed rollout counts inefficient, while skill metadata activates partially overlapping jury dimensions.

G Optimization Stability and Rollout Correction

Rollouts are generated by SGLang workers while log-probabilities and gradients are computed by Megatron workers, so the rollout policy π_{roll} and proximal training policy π_{prox} can differ even under periodic weight synchronization (Yao et al., 2025a; Zheng et al., 2025). We apply token-level truncated importance sampling (TIS) (Ionides, 2008; Yao et al., 2025a),

$$w_{i,t}^{\text{TIS}} = \min\left(\frac{\pi_{\text{prox}}(a_{i,t} \mid s_{i,t})}{\pi_{\text{roll}}(a_{i,t} \mid s_{i,t})}, C\right), \tag{7}$$

with cap $C = 2$. Separately, the policy ratio relative to π_{prox} is clipped asymmetrically,

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(a_{i,t} \mid s_{i,t})}{\pi_{\text{prox}}(a_{i,t} \mid s_{i,t})}, \quad \bar{\rho}_{i,t}(\theta) = \text{clip}_{[0.80,\,1.28]}(\rho_{i,t}(\theta)). \tag{8}$$

The TIS-corrected clipped surrogate and final objective are

$$\ell_{i,t}(\theta) = w_{i,t}^{\text{TIS}} \min\Big(\rho_{i,t}(\theta)\hat{A}_i,\ \bar{\rho}_{i,t}(\theta)\hat{A}_i\Big), \tag{9}$$

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{i,t}[\ell_{i,t}(\theta)] - \beta\,\text{KL}(\pi_\theta \,\|\, \pi_{\text{SFT}}). \tag{10}$$

TIS limits variance from rollout–training mismatch; policy-ratio clipping limits local update magnitude; and the KL anchor retains the SFT checkpoint as a behavioral reference. We track token entropy, clipped-token rate p_{clip}, and TIS-truncation rate p_{TIS} to separate entropy collapse, excessive local movement, and rollout mismatch.

H Public-Benchmark Consistency Check

We evaluate Base, SFT, and SFT+RL on BFCL (Patil et al., 2025) and ToolBench (Qin et al., 2024) subsets not used to select the proprietary RL mix. These labels are assigned after observing checkpoint behavior and are therefore structural analogues rather than prospective external validation.

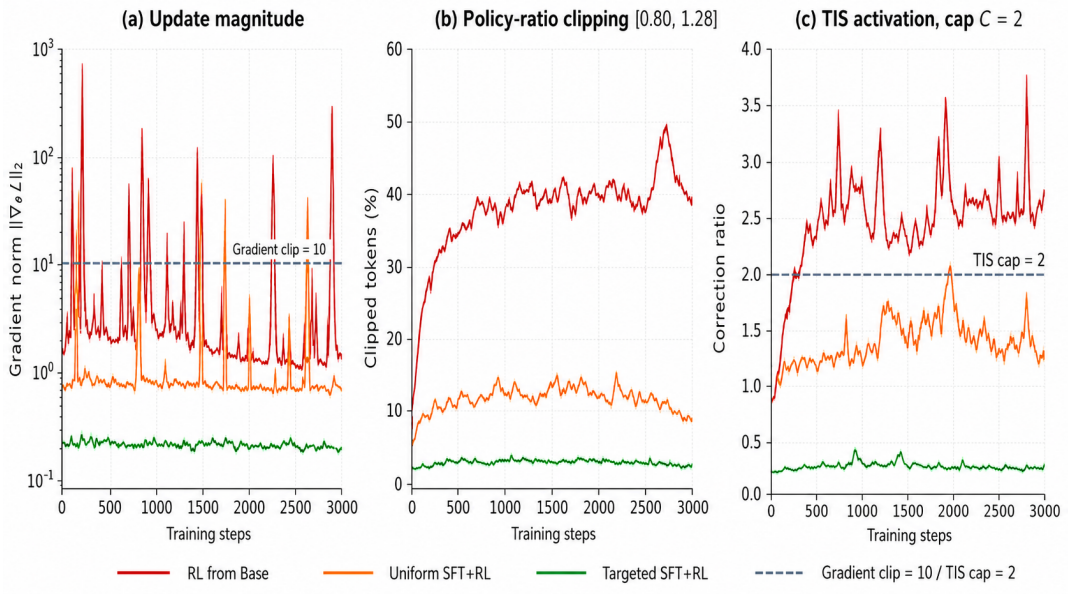


Figure 5: **Optimization stability and rollout correction.** Targeted SFT+RL exhibits smaller update magnitudes, lower clipping activity, and weaker TIS correction pressure than RL from Base or uniform SFT+RL. The gradient clipping threshold is 10; the policy-ratio interval is $[0.80, 1.28]$; TIS uses $C = 2$.

Observed pattern	Benchmark subset	Base	SFT	+RL	Δ_{RL}
Imitation-like	BFCL (Single-Tool)	82.1%	**93.4%**	92.1%	-1.3
Lift-like	ToolBench (Multi-Step)	74.2%	81.5%	**84.3%**	$+2.8$
Discovery-like	BFCL (Missing Params)	61.4%	68.2%	**81.5%**	$+13.3$

Table 8: External consistency check on public tool-use subsets.

BFCL Single-Tool has a narrow action space, and SFT learns syntax and selection convention before RL slightly regresses the calibrated mode. ToolBench Multi-Step admits multiple valid tool orderings; both SFT and RL add value. BFCL Missing Parameters requires recognizing absent required parameters rather than blindly emitting a call; RL provides the largest additional gain. This last pattern is a structural analogue of Discovery behavior, not a proxy for the proprietary non-disclosure task.

I Qualitative Case Study: Discovery-Regime Abstraction

To understand why RL can break through the SFT plateau on non-disclosure, we analyze sanitized trajectories where a tool returns competitor-sensitive bidding behavior alongside advertiser metrics.

SFT failure mode: imitation and leakage. The SFT model can remain factually responsive while exposing a restricted raw value:

> *"Your impression share dropped because Competitor X increased their top-of-search bid to $4.50, outbidding your maximum of $3.00."*

RL success mode: constrained abstraction. The targeted RL model instead produces an actionable abstraction:

> *"Your impression share dropped significantly due to increased competitive density at the top-of-search. To regain visibility, consider directionally increasing your top-of-search modifier while monitoring efficiency."*

This contrast illustrates the reward target but is not itself safety evidence; the independent SME audit in Appendix J provides the quantitative held-out validation.

J Jury Calibration and Independent Human Audit

Jury calibration. We calibrate the open-weight jury (DeepSeek-R1, Qwen3-235B-A22B) against a 200-trajectory human-rated seed set stratified across completed domain skills. Two SME raters independently

score each trajectory's <final_response> on a 5-point quality scale; inter-rater agreement is Cohen's $\kappa = 0.74$. Per-juror Spearman with averaged human scores is 0.76 (DeepSeek-R1) and 0.79 (Qwen3-235B-A22B); inter-juror Spearman is 0.81; ensemble jury versus averaged human is 0.78. The pre-specified deployment threshold is $\rho \geq 0.75$.

Independent non-disclosure audit. The safety audit is a separate held-out annotation set from the 200-trajectory calibration sample. SME-authored standard and adversarial prompts are used to evaluate direct/indirect leakage, factual correctness, actionability, excessive vagueness, refusal, and an overall acceptable-safety judgment. Two independent SME annotators achieve Cohen's $\kappa = 0.76$; disagreements are adjudicated by a second SME cohort.

Metric	Control	SFT	SFT+RL
Standard-prompt leakage rate	12.40%	11.80%	2.90%
Adversarial-prompt leakage rate	24.70%	22.90%	6.80%
Informative/actionable response rate	84.60%	86.20%	85.70%
Refusal rate	5.80%	5.50%	6.40%
Acceptable safety rate	76.90%	78.40%	91.80%

Table 9: Full independent SME audit of non-disclosure. The result separates leakage from usefulness and refusal behavior.

K Per-Skill SFT Trajectory Volumes and RL Mix

SFT trajectories are unevenly distributed across skill families, weighted by empirical difficulty and downstream importance. Exact per-skill training counts are proprietary; relative weights are summarized below. The RL mix re-weights toward exploration-heavy skills identified by the routing diagnostic and excludes Imitation-regime skills.

Regime	SFT mix weight	RL mix weight
Imitation	1.0×	0× (excluded)
Lift	1.0×	1.0×
Discovery	1.5×	2.5×
OOD (Toucan, regularizer)	0.3×	–

Table 10: Trajectory mix weights normalized within each stage relative to Lift at 1.0×.

L Hyperparameters and Empirical Sweeps

GRPO uses group size $G = 8$, KL coefficient $\beta = 0.04$ (constant; no annealing), reward weighting $\alpha = 0.4$ for R_{tool} and $1-\alpha = 0.6$ for R_{jury}, rollout context limit 64K tokens, and learning rate 5×10^{-7} with AdamW and linear warmup over 100 steps. We use asymmetric policy-ratio clipping with $\epsilon_{\text{low}} = 0.20$ and $\epsilon_{\text{high}} = 0.28$, giving $[0.80, 1.28]$. TIS uses $C = 2$. SFT uses learning rate 1×10^{-5}, batch size 64 trajectories, and three epochs.

The specific values for α, β, sequence limits, and data-mix ratios were selected following 83 large-scale development experiments, including unsuccessful RL-from-Base attempts. The exact sensitivity surfaces are proprietary. Recipe compute in Table 4 includes only the attributable SFT/RL compute of the retained recipes; shared hyperparameter-search and infrastructure bring-up compute are excluded from recipe totals because they are shared development costs rather than cleanly attributable to one recipe.

M Operational Metrics

The targeted candidate and Control were measured over the same traffic slice and serving configuration during a fixed online evaluation period. Exact traffic volume, evaluation dates, and shadow/live assignment mode cannot be disclosed under confidentiality requirements. Accordingly, the table reports normalized changes only and is interpreted as operational evidence rather than a reproducible randomized causal estimate of online impact.

Operational metric	Change vs. Control
Supported QPS	$+10\%$
Generation throughput	$+2.9\%$ tokens/s
P50 user-perceived latency	-4.2%
P95 user-perceived latency	-6.8%
Recommendation acceptance rate	$+3.6$ percentage points
Multi-turn task-completion rate	$+5.4$ percentage points
Recovery after a failed tool call	$+7.1$ percentage points
Average turns to resolution	-5%
Session abandonment rate	-2.1 percentage points

Table 11: Serving-period changes for the targeted candidate relative to Control. Negative latency, turns-to-resolution, and abandonment changes are improvements.

N Infrastructure for GPT-OSS Reinforcement Learning

The advertiser environment is built on a Megatron-backed (Shoeybi et al., 2019) SLIME (Zhu et al., 2025) stack with a disaggregated SGLang (Zheng et al., 2024) rollout engine. We extend this stack to support multi-turn beta-API rollouts, skill-aware reward shaping, heavy-tailed token lengths, and GPT-OSS-120B MoE synchronization. We describe framework-level changes; proprietary reward functions, datasets, and product-specific tool implementations are not disclosed.

Disaggregated RL setup. Training and inference run on disjoint GPU pools: while Megatron performs the optimizer step for batch k, SGLang generates rollouts for batch $k+1$ and reward workers score batch $k-1$. This asynchronous design hides rollout latency behind training compute. The training policy and rollout engine use different parallel decompositions and weight layouts, so each policy update requires explicit Megatron→SGLang synchronization.

SLIME: custom rollout generator and reward hub. SLIME dispatches inference to SGLang through a load-balancing router, computes rewards in a separate `RewardHub` actor, and returns batches to Megatron. We replace the default single-turn completion with a multi-turn `generate_rollout` loop implementing model action → tool dispatch → environment observation → next state. The generator dispatches structured JSON tool calls, parses outputs, appends observations, and masks tool-output tokens from the loss. The reward hub is extended for the multi-juror ensemble and skill-aware rubric routing.

MoE weight synchronization for disaggregated rollout. GPT-OSS uses fused, interleaved gate/up projections for experts, whereas Megatron emits unfused per-expert tensors. During synchronization, these tensors are repacked into the fused `w13/w2` layout served by SGLang. We add an expert-parallel-aware load path that materializes only local experts and restores the SwiGLU gate/up ordering. Under tensor parallelism, the all-gather that reconstructs the GLU layout must be applied consistently to weights and biases; extending the same re-chunking transformation to biases makes Megatron→SGLang transfer bit-correct.

Context parallelism for learnable-softmax attention. GPT-OSS combines learnable-softmax attention with interleaved full-attention and sliding-window layers. We add context-parallel support through a custom attention module. Full-attention layers use zigzag ring attention over sharded $q/k/v$ tensors; the learnable-softmax scale is threaded through forward and backward. Sliding-window layers instead all-gather $Q/K/V$, run local flash attention with the window mask, and re-slice outputs, because the global zigzag permutation breaks locality.

Heavy-tail handling and reward routing. We bucket rollouts by total context length and dynamically resize each batch to a fixed token budget, keeping memory approximately stable without truncating long trajectories or heavily padding short ones. Skill metadata travels with each rollout through generation and reward scoring so the reward hub can activate the appropriate rubric dimensions. Together, asynchronous rollout, MoE synchronization, context-parallel attention support, and token-budget batching make multi-node long-context GRPO feasible for GPT-OSS-120B.