
Small Language Models as Autonomous Operational Agents

Ahmed Elmokashfi
Amazon Web Services (AWS)
rahelmok@amazon.com

Usama Ahmed
Amazon Web Services (AWS)
ahmusama@amazon.com

Abstract

Small language models are usually cast as executors inside agentic systems: they call tools, follow procedures, and defer anything hard to a frontier model. We ask whether an 8B model can instead hold the *planning* role in a production incident-response pipeline, for a domain absent from public pretraining corpora. Our method combines continued pretraining over roughly 100,000 internal domain documents with Reinforcement Learning with Verifiable Rewards (RLVR) via GRPO against a reference-grounded judge, preceded by a diagnostic establishing that reward, rather than further imitation, is the instrument the model’s deficit calls for: across 337 incident reports the domain-adapted model’s deficit tracks how much it must infer rather than what it knows, and its shortfalls are failures of precision rather than direction. Trained on 9,400 expert trajectories and evaluated as the planner in a live workflow over 160 paired incidents, the model raises mean end-to-end quality from 5.72 to 6.53 out of 10 ($p = 1.1 \times 10^{-5}$) and lifts the share of incidents clearing the actionable bar from 37% to 60%. A planner ablation over 561 incidents postdating the training cutoff detects no difference between the 8B model, invoked once with retrieval withheld, and a retrieval-augmented frontier planner on cause identification and next-action accuracy ($p = 0.073$ and $p = 0.628$); these tests bound a difference rather than establish equivalence. Stratification localizes the residual deficit to one operation — resolving an aggregate descriptor into the elements it currently contains — which is dynamic state that does not belong in model weights. Our run settles at a KL divergence of ~ 0.10 without degenerating.

1 Introduction

Operational domains at scale run on expert diagnostic reasoning. A senior engineer confronted with a degraded system does not execute a fixed checklist; they form a hypothesis, choose the cheapest measurement that discriminates between competing explanations, and branch on the result. That capability takes months to acquire, is lost continuously to workforce attrition, and varies across practitioners. The same is true of medical triage, incident response, and compliance review.

Frontier-model agents can approximate this reasoning, but applying them to a domain outside their pretraining distribution remains awkward. A production agentic pipeline chains context fetching, knowledge-base lookup, skills loading, planning, tool calls, and synthesis — accumulating to tens or even hundreds of model invocations per incident. Cost scales with call count, latency accumulates, every stage is a failure mode, and the arrangement presumes connectivity to an external API, which rules out edge deployment, air-gapped environments, and settings where incident data cannot leave the operator’s boundary. Small language models invert each of these properties. The open question is whether an SLM can carry a multi-step agentic task itself, or whether it is confined to executing procedures that something larger wrote.

We investigate this in an out-of-distribution operational domain — network incident troubleshooting — combining continued pretraining, which instills domain knowledge, with RLVR, which reinforces troubleshooting planning procedures. Because the second stage is expensive and only sometimes appropriate, we treat the choice to apply it as an empirical question and answer it before training (§3.3). Reinforcement can sharpen behaviour a model already produces intermittently; it cannot install knowledge the model lacks, and distinguishing those two situations turns out to be measurable. Recent work has established that RLVR need not be restricted to domains with binary checkable answers [Gunjal et al., 2025, Dash et al., 2026, Su et al., 2025], and our reward design builds on that line. But all of it evaluates on static benchmarks with single-turn outputs, and none reports how training-time policy drift relates to deployment quality — the gap that turns out to matter most once the paradigm meets a live system.

Contributions.

1. **A readiness diagnostic for RLVR** that separates a knowledge deficit from an application deficit before training, over 337 internal incident reports (§3.3).
2. **Production-validated RLVR for agentic planning:** as the planning step of a live incident-response workflow over 160 paired incidents, the trained 8B model raises mean quality from 5.72 to 6.53 out of 10 and lifts the share of incidents clearing the actionable bar from 37% to 60% (§5).
3. **Retrieval-free planning at 8B:** over 561 incidents postdating the training cutoff, we detect no difference between the model with retrieval withheld and a retrieval-augmented frontier planner on either primary axis, and stratification localizes its residual deficit to entity resolution rather than reasoning (§6).
4. **A training-dynamics account for novel output formats**, arguing that KL magnitude is not a safety signal when the reference policy cannot produce the target format at all (§4).

2 Related Work

Domain adaptation and SLM agents. Continued pretraining on domain corpora is the standard method for specializing general-purpose LLMs [Gururangan et al., 2020], and Yang et al. [2025b] show narrow corpora suffice when augmented with synthetic diversity; our first stage (§3.2) follows this recipe. Work on small models as agents has concentrated on tool use — teaching compact models to emit well-formed API calls [Schick et al., 2023, Liu et al., 2025] — and on routing architectures that defer difficult inputs to a larger model [Varangot-Reille et al., 2026], with a recent position paper arguing SLMs are inherently better suited to repetitive agentic invocations [Belcak et al., 2025]. Closer to our setting, a line of work already treats compact models as planners rather than executors. Zhang et al. [2024] decompose constrained tasks hierarchically across collaborating agents and report that an 8B model can serve as the planning core; Rawat et al. [2025] fine-tune smaller models to emit multi-step plans that are refined after each tool result; and Zhang et al. [2026] reach a conclusion opposite to ours in a neighbouring domain, externalizing domain knowledge into verifiable tools rather than model parameters. Our contribution is therefore neither the idea of an SLM planner nor the claim that internalizing domain knowledge is always preferable. It is an evaluation of one such planner inside a production incident-response workflow, and a boundary on what is worth internalizing: stable troubleshooting knowledge transfers into weights, while the entity mapping of §6 does not. Our planner also emits a complete plan in one pass rather than revising it after each observation, which §5 identifies as a limitation on cascading failures.

RLVR beyond binary verification. DeepSeek-R1 [Guo et al., 2025] established RLVR for domains with binary checkable ground truth. Later work extends it to soft rewards through rubrics [Gunjal et al., 2025], atomic requirement checklists that also exhibit reward inflation under self-verification [Dash et al., 2026], cross-domain generative verifiers [Su et al., 2025], and rubric anchoring [Huang et al., 2025]; orthogonally, GRPO [Shao et al., 2024] has been extended to multi-turn settings with turn-level credit assignment [Wei et al., 2026, Shan et al., 2026]. All target static benchmarks with single-turn outputs. We differ in two ways: our output is a structured multi-step plan scored holistically rather than per step, and deployment surfaces training dynamics (§4) that benchmark evaluation cannot reveal, including a stable operating point at a KL divergence conventional heuristics would reject. Appendix H positions this work against the three closest of these papers point by point.

3 Method

3.1 Task formulation

The model receives a natural-language incident description and emits a structured JSON investigation trajectory in one inference pass. Each step contains four fields: a *reasoning* field stating the diagnostic hypothesis under test, the *diagnostic tool* selected to test it, the *anticipated observation*, and a *branching decision* conditioned on what is actually observed. Downstream tool-calling agents execute the plan step by step.

This requires two capabilities a general-purpose LLM lacks. The model must *know* the domain — internal system names, tool APIs, failure-mode signatures, escalation protocols — none of which appear in public pretraining corpora, which domain adaptation (§3.2) supplies. It must also *reason* over that knowledge, since a model that knows every runbook by heart may still investigate in the wrong order or miss a mandatory safety check; RLVR (§3.4) closes that second gap. Structured output serves both stages: it decouples planning from execution, makes a plan auditable before any action is taken, and renders format itself partially verifiable, which the reward exploits.

3.2 Domain adaptation

We start from Qwen3-8B [Yang et al., 2025a], an 8.2B-parameter dense transformer, and run full-parameter continued pretraining over approximately 100,000 domain records — about 500M tokens drawn from internal wikis, incident reports, and code repositories — for roughly 19.5 hours. Training and evaluation loss converge closely, which is consistent with fit rather than memorization though it does not establish the absence of memorization. Appendix A gives the collection pipeline and configuration.

The resulting model internalizes system vocabulary, tooling names, and failure-mode structure, answering factual domain questions at an 82% non-wrong rate over 1,500 questions. Asked instead to reason about operational issues, it produces a helpful answer — one listing correct troubleshooting steps together with the diagnostic tools needed — in only 32% of 300 cases. Knowledge transferred; the ability to act on it did not. These two rates establish that a gap exists but do not locate it, since the benchmarks differ in both format and difficulty. The argument of §3.3 rests instead on the rubric-scored comparison, where both families are drawn from one incident corpus and scored by one instrument, so the gap there is attributable to inference demand alone.

3.3 Does the domain-adapted model know enough to reinforce?

RLVR is worth applying only to a model that already holds the relevant knowledge and fails at applying it: reward can sharpen behaviour a model produces intermittently, but cannot install knowledge it lacks. To establish which case we were in, we built two benchmark families from the same corpus of 337 internal incident reports. *Writing* items give the model the incident’s facts and ask for an operational artifact; *troubleshooting* items give it raw observations, alarms, and response timelines, and require it to work out what happened and what to do. Both demand domain knowledge the model must supply itself and differ only in whether the incident’s facts are given. Each item carries an expert-written rubric whose criteria a judge marks satisfied, partially satisfied, or unmet; Appendix B gives the full protocol and the per-subtype scores.

The model reaches 60.2% on writing against 46.2% on troubleshooting, and writing is stronger on 73% of individual incidents. The informative pattern is *within* that gap. Ordering the five subtypes by how much the model must work out for itself — summaries compress supplied facts, runbooks generalize a scenario into a procedure, recommendations judge what should have been done differently, and troubleshooting derives the facts at all — produces a gradient that runs continuously across the two families rather than between them: the spread within writing is 19 points, while the step from the hardest writing subtype to the easiest troubleshooting subtype is under 3 (Table 3 in Appendix B). Difficulty therefore does not track task family, output format, or rubric style, all of which change at that boundary; it tracks inference demand, which varies smoothly across it.

This identifies the deficit as one of application rather than knowledge, which is the condition under which reward, rather than further imitation, is the instrument to reach for. Two further conditions hold. Reward is reachable rather than sparse: 67.9% of troubleshooting criteria earn at least partial credit.

And the residual failure is verifiable — answers fall short on precision rather than direction, with 59% of partially satisfied criteria marked down for omitting a specific value, threshold, or component against 1% for addressing the wrong aspect. Whether a plan names a threshold, and whether that threshold is correct, can be checked mechanically in a way the quality of its reasoning cannot.

Two properties of this setting bear on the choice of reinforcement over supervised imitation. Our reward credits alternative investigation orders that reach the same mandatory decision points, and the deficit above lies in precision rather than direction. Neither is expressible by a likelihood objective over single reference trajectories, which maximizes agreement with one ordering and weights every token of it alike. We do not, however, measure how much of the production gain of \$5 supervised specialization on the same trajectories would recover. Within training the format component is separable from the rest: the gate is capped at 0.25 and saturates within the first 9% of steps, after which all reward improvement is in the judge score, raw 0.52 to 0.73.

3.4 RLVR training

Data and optimization. We reconstructed 9,400 expert trajectories from resolved incidents and operational runbooks using a frontier-model agent with read access to the relevant diagnostic systems, then reviewed and corrected them with human domain experts. Two problems dominated this work: many resolved incidents record the outcome but not the diagnostic chain that produced it, and branching coverage requires deliberately seeking out edge-case paths that rarely appear in resolved tickets. We optimize with Group Relative Policy Optimization (GRPO) [Shao et al., 2024], group size 8, a cosine learning-rate schedule, and a KL penalty against the domain-adapted reference policy. GRPO is preferable to PPO [Schulman et al., 2017] here because it eliminates the critic network, and because group-relative advantage suits a task with multiple valid solutions: when several sampled trajectories investigate an incident in different but equally sound orders, group-relative scoring assigns them comparable advantage instead of forcing a value function to treat one ordering as canonical.

Reward. Our reward differs from the literature in what it is grounded against. Rubrics as Rewards applies domain-generic rubrics uniformly; Soft-SVeRL decomposes each prompt into atomic requirements without reference to any solution. Ours is *reference-grounded*: Claude Sonnet 4.5 scores each generated trajectory against an expert reference trajectory for the same incident. This is what makes partial credit meaningful in an agentic setting. An alternative investigation order reaching the same mandatory decision points is credited, while one that skips a mandatory check or violates a safety constraint is penalized — and the judge can tell the difference only because it can see what a correct trajectory for *this* incident looks like. A format compliance gate worth 0.25 points bootstraps structured output, and a judge reward worth up to 1.0 points scores seven weighted dimensions, listed with their criteria in Appendix C and weighted so that process-quality dimensions carry more than content-recall dimensions, because content is what domain adaptation already installed. Raw judge scores are shaped as

$$r_{\text{judge}} = \max\left(0, \frac{s_{\text{raw}} - 0.4}{0.6}\right), \quad (1)$$

which discards the bottom 40% of the judge’s range — trajectories with no usable structure — and rescales the remainder. The practical ceiling is approximately 0.80 rather than 1.0, which is principled rather than a calibration failure: the dimensions are in tension, and expert references do not simultaneously maximize all of them either, since a conciseness-maximizing plan is not an order-optimal plan.

4 Training Dynamics and What KL Measures

We train for one epoch on the full dataset, monitoring judge reward, KL divergence, completion length, and format compliance. Three phases emerge, we present the plots and per-phase detail in Appendix D. *Format discovery* consumes the first ~9% of steps, during which judge reward stays near zero while format compliance saturates: the domain-adapted policy generates free text, but training requires structured JSON. *Content learning* follows to roughly epoch 0.50, with judge reward climbing from 0.20 to 0.55 shaped (0.52 to 0.73 raw). *Plateau* holds thereafter at 0.55 shaped — 91% of the expert ceiling — with KL stable near 0.10 and none of the runaway length growth characteristic of verbosity reward hacking. The plateau is not training failure: per-example performance is strongly

Table 1: End-to-end production quality on 160 paired incidents, scored 1–10 against actual resolutions.

	Mean	≥ 7	≥ 8
DA baseline	5.72	36.9%	20.0%
RLVR model	6.53	60.0%	43.8%

bimodal, scoring ≥ 0.7 on 68% of examples and ≤ 0.2 on 14%, so the ceiling is set by coverage gaps rather than by an inability to learn patterns that are present.

Our run stabilizes at a KL of 0.10 against the domain-adapted reference, five times the 0.02 monitoring threshold we set going in. That threshold was conservative by construction: values of this order come from per-update trust-region practice, which bounds the divergence between consecutive policy iterates, whereas reference-KL accumulates over an entire run and is governed by no comparable convention. By our own trigger the model should have been degenerate. It is not: evaluated on 160 paired production incidents in the next section, it outperforms the reference policy by a wide margin. We read the large KL not as pathological drift but as evidence that the model acquired a capability the reference policy did not have, since moving from knowledge retrieval to procedural planning requires reorganizing the output distribution. The generalizable claim is that KL measures distance from the reference policy, not from quality: when that policy cannot produce the target output at all, large KL is a necessary condition for successful training rather than a warning sign, and no fixed threshold on its magnitude is informative. What matters is the joint trajectory of KL and reward. Both rising indicates learning; KL rising against flat or falling reward indicates collapse.

5 Production Evaluation

To test whether offline reward translates into deployment quality, we integrated the trained model into a production incident-response workflow as its *planning* step: it receives parsed incident context and produces a data-gathering plan specifying which telemetry sources to query, with what parameters, over what time windows. Downstream agents execute the plan using 20 operational tools and feed results to a frontier-model synthesis step that writes the final analysis. Because all variants share that downstream pipeline, differences in end-to-end quality trace to plan quality alone. We evaluate on 160 real resolved incidents spanning multiple network infrastructure types, each run with both the RLVR-trained model and the domain-adapted baseline as planner, which pairs the comparison and controls for incident difficulty. An LLM evaluator, Claude Opus 5, scores each final synthesis from 1 to 10 against the actual resolution, assessing problem identification, root cause, status assessment, actionability, and absence of fabricated entities. The reward judge of §3.4 is therefore not the evaluator, and the evaluator does not score plans directly: it scores the synthesis produced downstream from executed tool results. The 160 incidents are held out from both the pretraining corpus and trajectory construction. We report no human calibration of the 7 and 8 thresholds, so those labels rest on the evaluator’s judgement rather than on expert assessment.

The trained model improves mean score by 0.81 points (Wilcoxon signed-rank $p = 1.1 \times 10^{-5}$, rank-biserial $r = 0.43$; Table 1). At the ≥ 7 bar — above which plans are actionable without human correction — it covers 60% of incidents against 37% for the baseline, and at ≥ 8 , where plans match expert quality, the gap widens to 24 points. Head-to-head it wins 61% of incidents, ties 13%, and loses 26%. The advantage is non-linear across thresholds: for ≥ 5 the gap is only 6 points, since both models handle straightforward incidents adequately, and it peaks at ≥ 8 . RLVR’s benefit is concentrated in enabling high-quality plans rather than in marginally improving poor ones. Offline reward predicted this ranking correctly, which is not guaranteed: reward hacking can produce offline scores that do not transfer. Our reference-grounded judge appears well calibrated, plausibly because it evaluates substantive correctness rather than surface plausibility.

Deployment safeguards and data handling. All diagnostic tools exposed to the planner are read-only; the system just produces recommendations. Incident data processed by external models is handled under existing internal data-handling authorization and access controls. We report no measured latency, cost, or memory, and the deployed pipeline retains a frontier model for synthesis, so these results do not describe a fully local system.

Table 2: Ablation over 561 held-out incidents, stratified by whether the record names a network element (11% non-conforming residual omitted). Cells are SLM / frontier, in percent; head-to-head remainders are ties. * is the only primary-axis difference reaching significance ($p = 0.007$).

Incident scope	Share	Cause	Next action	Contradicts	Head-to-head
Element-scoped	68%	18.1 / 20.4	20.4 / 18.3	7.6 / 10.2	30.9 / 32.2
Grouping-scoped	21%	11.9 / 15.3	7.6 / 19.5 *	38.1 / 16.1	15.3 / 39.0

Where it fails. The trained model scores below 7 on 40% of incidents, and the failures are not uniformly distributed. Multi-system cascading failures dominate: the causal chain spans more infrastructure layers than any single training trajectory demonstrates, so the model investigates one layer correctly without tracing cross-layer causation. Training-distribution gaps follow, which yield to targeted data augmentation. Two smaller categories are more instructive. Monitor-and-wait incidents are cases where the correct action is inaction, and no dimension of our reward credits correctly declining to investigate, so the model always produces a plan; missing organizational context covers resolutions depending on facts absent from the incident record. The lesson is to separate model limitations, task limitations, and reward-design limitations: the third is a systematic blind spot of any reward built around positive action, and practitioners should audit whether their reward has a path to full score for every valid behaviour, inaction included. Appendix E gives the category shares and the score distribution.

6 Planner Ablation Against a Frontier Model

We ran the deployed pipeline in two configurations over the same incidents, varying only the planner. The *frontier* planner is a frontier model with access to operational runbooks, a systems knowledge base, and several planning calls; the *SLM* planner is the 8B model of §3.4, invoked once with retrieval withheld by design: the ablation asks how far fine-tuning can substitute for retrieval and frontier-scale planning. Telemetry agents, their tools, and the synthesis stage are identical across configurations, which we verified by inventory diff (Appendix F). All 561 evaluation incidents postdate the training cutoff, making this a temporal holdout at the incident level: the SLM may transfer failure modes and remediation patterns from historical incidents but cannot have observed these incidents or their resolutions. Scoring uses the same evaluator as §5 but a per-axis rubric rather than a single score: each axis below is judged as a binary outcome against the documented resolution, and the head-to-head verdict is recomputed from those judgements (Appendix F). Paired differences are tested by exact McNemar; a provenance-blinded second judge reproduced the ordering at $\kappa = 0.672$.

Withholding retrieval produces no statistically detectable degradation on the primary axes. Exact cause identification is 15.3% for the SLM planner against 19.1% for the frontier planner ($p = 0.073$); next-action accuracy is 16.6% and 17.8% ($p = 0.628$). Head-to-head, the SLM wins 26.4% of incidents, the frontier planner wins 34.0%, and 39.6% tie. We do not read failure to reject as equivalence — the sample bounds the difference we could detect — but the aggregate null motivates asking where the systems diverge.

The SLM deficit concentrates in under-specified incidents. Performance separates sharply according to whether the incident record names a network element (Table 2). When an element is named, the two planners are similar on both primary axes and the SLM contradicts the documented human conclusion less often. When the incident is specified only by an endpoint pair or a logical grouping, the SLM’s contradiction rate rises roughly fivefold and its next-action accuracy falls substantially.

Inspecting these failures reveals a consistent mechanism. Without an element identifier, the SLM must resolve an aggregate descriptor into the elements to which subsequent telemetry queries should be directed, and without retrieval it frequently selects the wrong ones. Downstream agents then surface telemetry from an unrelated element, after which synthesis reasons coherently about the wrong subject. The missing operation is therefore not a troubleshooting inference but an inventory lookup: mapping a logical or aggregate descriptor to its current constituents. That mapping changes as the network changes, which makes it poorly suited to encoding in model weights, and it is precisely what the systems knowledge base supplies to the frontier planner.

The two planners are consequently close in aggregate but not interchangeable. The SLM’s characteristic strength is scope discipline: given an element-local symptom it keeps the investigation local rather than embedding it in an unsupported network-wide narrative, and it can correctly recommend no action for self-clearing incidents. Its characteristic failure is the mirror image, and the frontier planner is more robust to specification uncertainty because retrieval supplies the context needed to resolve groupings into entities; it also retains advantages on two secondary axes under stratification (Appendix F). The boundary is therefore not one of model scale but of what can usefully be internalized: the 8B planner transfers stable troubleshooting knowledge — failure modes, diagnostic strategies, remediation patterns — with no retrieval infrastructure, while its deficit is confined to a frequently changing entity mapping that should not live in model weights at all. This argues for a targeted hybrid rather than a larger planner: keep the 8B model for planning and invoke retrieval only for entity resolution.

7 Conclusion

An 8B model can hold the planning role in a production incident-response pipeline for a domain it never saw in pretraining, with two qualifications we expect to generalize. First, the sequence matters more than either stage. Continued pretraining installed the domain but not the ability to act in it — 82% on factual questions against 32% helpful troubleshooting guidance — and RLVR closed the second gap, not the first. The diagnostic told us in advance which gap was available, because the failures were of precision rather than direction. Second, the boundary we hit was not about model scale: with retrieval withheld we detect no difference between the 8B planner and a retrieval-augmented frontier planner on either primary axis across 561 held-out incidents, and where it fails it fails on one operation — mapping an aggregate descriptor to the elements it currently contains — which changes as the network changes.

Four limitations bound these claims. All results come from one operational domain, so the transfer argument rests on the method’s structure rather than on evidence; Appendix G sets out the conditions we believe a domain must meet. The ablation result is a failure to reject, not a demonstration of equivalence. Quality is scored throughout by LLM judges against documented human resolutions, with no human calibration of the thresholds we call actionable and expert-quality. We also do not measure what supervised fine-tuning on the same trajectories would recover, so the gain over the domain-adapted baseline is not attributed to reinforcement specifically. And the model still falls below the actionable bar on 40% of incidents, dominated by multi-system cascades: a coverage deficit rather than a capability one, and the case for specialized planners over adjacent subsystems.

References

- Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*, 2025.
- Saurabh Dash, Pierre Clavier, John Dang, Matthias Galle, Marzieh Fadaee, Ahmet Üstün, and Beyza Ermiş. Soft-sverl: Self-verified reinforcement learning with soft rewards, 2026. URL <https://arxiv.org/abs/2605.28561>.
- Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains, 2025. URL <https://arxiv.org/abs/2507.17746>.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL <https://aclanthology.org/2020.acl-main.740/>.

- Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, Jianguo Li, and Junbo Zhao. Reinforcement learning with rubric anchors, 2025. URL <https://arxiv.org/abs/2508.12790>.
- Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. Toolace: Winning the points of llm function calling. In *International Conference on Learning Representations (ICLR)*, 2025. URL <https://openreview.net/forum?id=8EB8k6DdCU>.
- Mrinal Rawat, Ambuje Gupta, Rushil Goomer, Alessandro Di Bari, Neha Gupta, and Roberto Pieraccini. Pre-act: Multi-step planning and reasoning improves acting in llm agents, 2025. URL <https://arxiv.org/abs/2505.09970>.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, volume 36, pages 68539–68551, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- Yingyu Shan, Yuhang Guo, Zihao Cheng, Zeming Liu, Xiangrong Zhu, Xinyi Wang, Jiashu Yao, Wei Lin, Hongru Wang, and Heyan Huang. Learning from own solutions: Self-conditioned credit assignment for reinforcement learning with verifiable rewards, 2026. URL <https://arxiv.org/abs/2606.18810>.
- 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>.
- Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains, 2025. URL <https://arxiv.org/abs/2503.23829>.
- Clovis Varangot-Reille, Christophe Bouvard, Antoine Gourru, Mathieu Ciancone, Marion Schaeffer, and François Jacquenet. Doing more with less: A survey on routing strategies for resource optimisation in large language model-based systems. *Journal of Artificial Intelligence Research*, 86:19801, 2026. doi: 10.1613/jair.1.19801. URL <https://acm.org>.
- Quan Wei, Siliang Zeng, Chenliang Li, Zhongruo Wang, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via fine-grained reward structure and credit assignment, 2026. URL <https://arxiv.org/abs/2505.11821>.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL <https://arxiv.org/abs/2505.09388>.
- Zitong Yang, Neil Band, Shuangping Li, Emmanuel J. Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL <https://openreview.net/forum?id=07yvxWDS1a>.

Cong Zhang, Derrick Goh Xin Deik, Dexun Li, Hao Zhang, and Yong Liu. Planning with multi-constraints via collaborative language agents, 2024. URL <https://arxiv.org/abs/2405.16510>.

Yongqiang Zhang, Mustafa A. Kishk, and Mohamed-Slim Alouini. Small models, big impact: Tool-augmented ai agents for wireless network planning. 2026. URL <https://arxiv.org/abs/2601.13843>.

A Domain adaptation: corpus and training configuration

A multi-source pipeline collected $O(100,000)$ training records — about 500M tokens under the Qwen tokenizer — from internal wikis, incident reports, and code repositories. The code-documentation path is worth describing: source packages were cloned, statically analyzed for interfaces and error-handling patterns, then fed to an orchestrated generation session that produced structured handbook documents, each validated against the actual codebase before inclusion.

We performed full-parameter continued pretraining on a single `ml.p4de.24xlarge` instance with ZeRO-2 sharding for approximately 19.5 hours, covering 2 epochs. The model reached 90.10% training accuracy at a final loss of 0.4205, with evaluation closely matching at 89.01% accuracy and 0.48 loss. The narrow train–eval gap is consistent with the corpus being absorbed rather than memorized, but does not by itself establish that.

Two evaluations characterize the resulting model. On a benchmark of 1,500 domain-specific factual questions it achieves an 82% non-wrong rate. On a separate benchmark of 300 questions asking for troubleshooting guidance, it produces a helpful answer (i.e. listing correct troubleshooting steps together with the diagnostic tools needed) in only 32% of cases. This split is what motivated the readiness diagnostic of §3.3.

B Readiness diagnostic: protocol and supporting evidence

Construction. Both benchmark families are built from the same corpus of 337 internal network incident reports, which keeps domain and difficulty distribution fixed across the comparison. *Writing* items supply the incident’s facts and request an operational artifact; *troubleshooting* items supply raw observations, alarms, and response timelines, and require the model to infer what happened. The families are therefore matched on knowledge demand and differ in inference demand.

Knowledge is genuinely required. Only 5% of writing criteria can be satisfied by restating the prompt, and 51% require content largely absent from it. Neither family can be answered by paraphrase.

Scoring. Every item carries an expert-written rubric of criteria, each naming one specific point a correct answer must make. A judge model marks each criterion satisfied, partially satisfied, or unmet, treating vague content as unmet. An item scores the fraction of criteria satisfied, with half credit for partial.

Table 3: Rubric score by subtype, ordered by how much the model must infer. Domain knowledge is required throughout; only the availability of the incident’s facts varies. The ordering runs continuously across the two task families rather than between them.

Task	Subtype	<i>n</i>	Score
Writing	Executive summary	318	69.2%
Writing	Operational runbook	316	57.5%
Writing	Improvement recommendations	205	50.2%
Troubleshooting	Diagnosis from observations	329	47.3%
Troubleshooting	Timeline review and process improvement	324	45.0%

Reward reachability. 67.9% of troubleshooting criteria earn at least partial credit, and the root cause is at least partially identified in 76.8% of items. Exploration therefore begins from a signal rather than having to discover one, which is a precondition for GRPO to make progress at all.

Failures are of precision, not direction. Among partially satisfied criteria, 59% are marked down for omitting a specific value, threshold, or component, against 1% for addressing the wrong aspect. The characteristic error is writing “monitoring for hardware resource exhaustion with alerts” where the criterion requires “SRAM and TCAM monitoring with alerting at X%.” Whether a plan names a threshold, and whether that threshold is correct, can be checked mechanically in a way the quality of its reasoning cannot — which makes this a tractable reward target, provided credit is withheld unless the specific claim is right.

C Judge reward dimensions

Table 4: Judge reward dimensions. Process-quality dimensions are weighted above content-recall dimensions because domain adaptation already covers content; RL signal is spent on what it cannot teach.

Dimension	Weight	Scoring criterion
Investigation order	20%	Are cheap, high-discrimination measurements taken first?
Triage precision	20%	Is the initial severity and scope assessment correct?
Branch correctness	20%	Are branching decisions conditioned on the right observations?
Reasoning quality	15%	Does each step state a hypothesis the tool call can falsify?
Hard rule compliance	10%	Are mandatory checks and safety constraints respected?
Remediation specificity	10%	Are recommended actions concrete enough to execute?
Conciseness	5%	Is the plan free of steps that cannot change the decision?

Table 4 lists the dimensions and their weights. The format compliance gate is worth a further 0.25 points, and its asymmetry is deliberate: large enough to establish valid JSON quickly, small enough that a model cannot coast on formatting, and enough to keep gradient signal alive while content quality is still near zero. We validated the judge by having it score reworded versions of the reference answers.

D Training dynamics: curves, per-phase detail, and stopping criterion

Figure 1 plots all four monitored metrics over the run.

Phase 1, format discovery (epochs 0–0.09). Format compliance rises from 0.0 to near-saturation while judge reward remains near zero. Completion length is high and variable (>800 tokens) as the model explores the output space before converging on the rewarded structure. Practitioners should not interpret the initial KL spike as instability: when the reference policy genuinely cannot produce the target format, drift *is* the learning. The diagnostic signature is format reward rising while judge reward stays flat. If both remain flat, either the format reward signal is too weak or the format specification is too complex to discover through exploration.

Phase 2, content learning (epochs 0.10–0.50). Judge reward climbs from 0.20 to 0.55 shaped (0.52 to 0.73 raw). Completion length compresses to 550–650 tokens as the model allocates tokens to content rather than format exploration. KL continues rising but at a decelerating rate, reaching approximately 0.08. This phase is where training-data diversity has its primary effect: each unique pattern contributes to the reward gradient while repeated patterns contribute diminishing signal. Doubling unique patterns from 2,030 to 4,700 produced a 73% improvement in judge reward, while additional epochs over the same patterns produced negligible gains.

Phase 3, plateau (epochs 0.50–1.0). Reward holds at approximately 0.55 shaped (0.73 raw, 91% of the expert ceiling), KL stabilizes near 0.10, and format compliance holds at 0.23–0.24. Completion length drifts upward to 650–750 tokens without the runaway growth characteristic of verbosity reward hacking. Per-example analysis is strongly bimodal: ≥ 0.7 on 68% of examples, ≤ 0.2 on 14%. The ceiling is driven by coverage gaps rather than capacity limits.

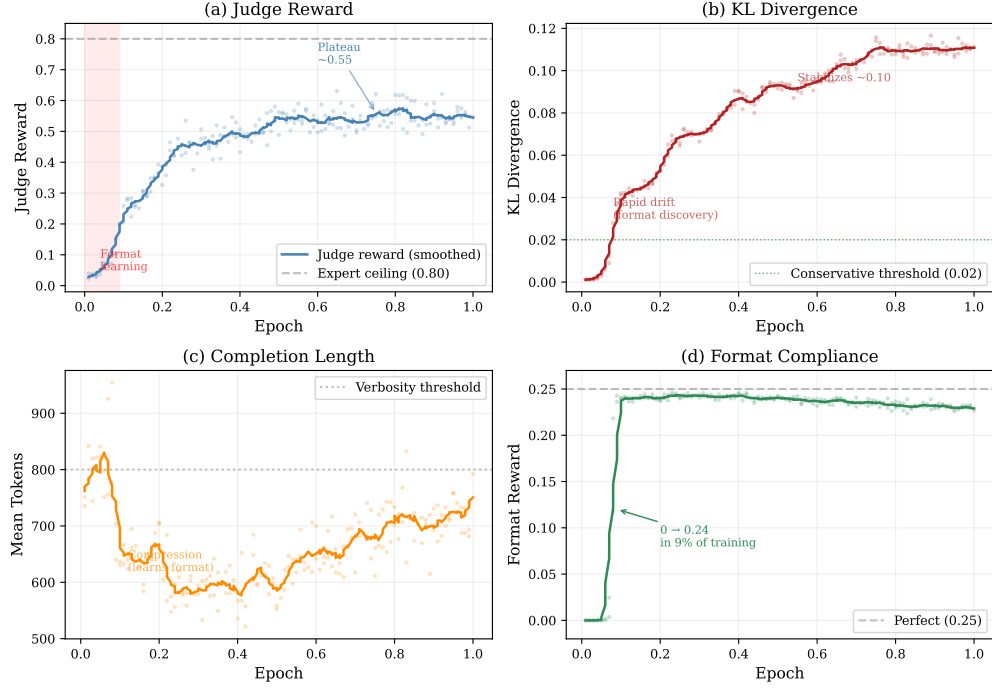


Figure 1: Training dynamics over one epoch of GRPO on 9,400 trajectories. (a) Judge reward rises through format discovery (shaded), content learning, and plateau. (b) KL stabilizes at ~ 0.10 , above conventional thresholds but non-pathological. (c) Completion length compresses once format is learned. (d) Format compliance is acquired within 9% of training.

Stopping criterion. We recommend halting when reward improvement falls below 0.02 over 0.2 epochs with KL variance under 10% and length variance under 15%. Continuing past this point risks reward hacking without improving deployment quality. The correct response to a plateau of this shape is more data, not more epochs.

E Failure taxonomy and score distribution

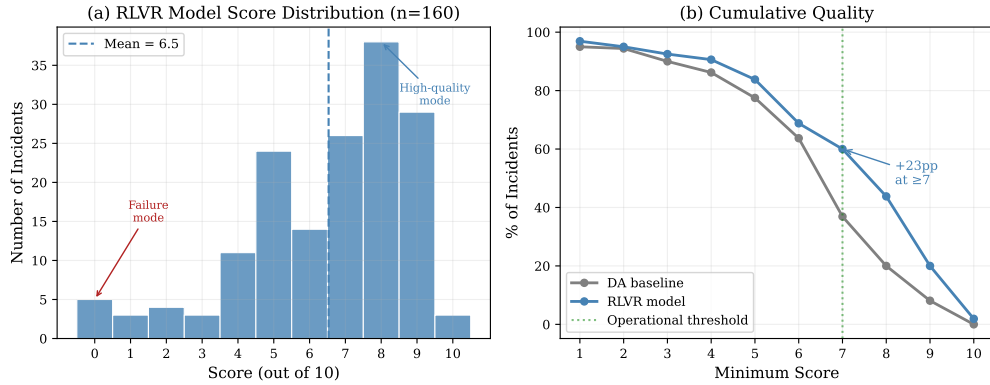


Figure 2: Production evaluation on 160 paired incidents. (a) The RLVR score distribution is bimodal — a mode at 8 and a failure tail at 0–3 — which the mean obscures. (b) The advantage over the baseline grows with threshold stringency: +23pp at the ≥ 7 bar and +24pp at ≥ 8 .

Figure 2 shows the score distribution and the threshold curve behind the headline numbers of §5. The trained model scores below 7 on 40% of production incidents, and we classify these failures as follows.

1. **Multi-system cascading failures (42%).** Incidents spanning multiple infrastructure layers, where the causal chain exceeds what any single training trajectory demonstrates. The model investigates one layer correctly but does not trace cross-layer causation.
2. **Training distribution gaps (27%).** Incident formats or alarm types absent from training data, producing generic or malformed plans. Addressable with targeted data augmentation.
3. **Monitor-and-wait incidents (9%).** Cases where the correct action is inaction — transient alarms, false positives. No reward dimension credits correctly declining to investigate, so the model always produces an investigation plan.
4. **Missing organizational context (6%).** Resolution requires knowledge absent from the incident description: ongoing maintenance windows, known degraded states, prior coordination.

Post-deployment failure analysis should distinguish model limitations (addressable through training), task limitations (inherent ambiguity in the input), and reward-design limitations (correct behaviour the reward does not credit). The third category — here, correctly doing nothing — is a systematic blind spot of reward functions designed around positive action.

F Ablation controls and secondary axes

Pipeline equivalence. Telemetry agents, their tools, and the synthesis stage are identical across the two configurations. We verified this by diffing every agent’s declared tool inventory between runs and excluding domains in which those inventories differed, so that the comparison varies planning alone.

Verdict computation. The head-to-head verdict is recomputed deterministically from the per-axis judgements rather than taken from the judge’s stated overall preference, and reported ties are derived from per-axis agreement rather than from an expressed indifference. A provenance-blinded second judge reproduced the ordering with substantial agreement ($\kappa = 0.672$).

Secondary axes. Under stratification the frontier planner retains advantages on two secondary axes: surfacing the artifact that justified closure ($p = 0.0019$) and evidential grounding ($p = 0.0034$). Both are capabilities directly supported by access to operational runbooks and the systems knowledge base, which is consistent with the entity-resolution account in §6 rather than with a general reasoning deficit.

Residual stratum. An 11% residual class of records does not conform to either scope category and is omitted from Table 2.

G When this method transfers

Four conditions appear necessary for a domain to support this approach. First, expert trajectories must be reconstructable from historical resolved cases; without them there is nothing to ground the reward against. Second, failure modes must be identifiable clearly enough to separate wrong approaches from merely suboptimal ones — a reward that cannot make that distinction collapses into a style score. Third, quality must decompose into independently scorable criteria. Fourth, the domain must tolerate multiple valid solution paths, since otherwise binary verification is available and simpler.

Three domains satisfy all four. Medical triage admits multiple valid examination orders, has mandatory differential checkpoints, and has hard contraindications that function much like our safety rules. DevOps incident response has substitutable observability queries and mandatory escalation conditions. Legal compliance review has flexible evidence ordering with mandatory statutory considerations. In each case the binding constraint is likely the first condition: reference trajectory reconstruction required a frontier-model agent with system access plus expert review, and is the most expensive part of the pipeline to reproduce.

H Positioning against concurrent work

Table 5 states our position against the three primary concurrent works on RLVR beyond binary verification. The reward-design row is where we are closest to prior work; the two evaluation rows are where we differ most. We want to be explicit about the implication: the soft-verifiable framing is not ours to claim. Rubrics as Rewards, Soft-SVeRL, and Crossing the Reward Bridge establish that RLVR extends beyond binary verification. What we add is evidence that it survives deployment, and a training-dynamics finding — the stable high-KL operating point of §4 — that benchmark evaluation structurally cannot produce.

Table 5: Positioning against the primary concurrent RLVR-beyond-binary literature. Bold entries mark properties unique to this work.

Dimension	This work	Rubrics as Rewards	Soft-SVeRL	Crossing the Bridge
Evaluation setting	Production, 160 paired incidents	Benchmark QA (medical, science)	Instruction-following benchmarks	5 academic domain datasets
Task type	Multi-step agentic planning	Single-turn QA	Single-turn instruction following	Single-turn QA
Model size	8B (Qwen3-8B)	Multiple sizes	Multiple sizes	Multiple sizes
Reward design	Reference-grounded 7-dimension judge	Domain-generic rubrics	Atomic requirement checklists	Learned cross-domain verifiers
Production validation	Yes, live workflow	No	No	No
Agentic planning	Yes, structured trajectory	No	No	No
Retrieval-free generalization	Yes, 561-incident temporal holdout	Not reported	Not reported	Not reported
KL drift analysis	Yes, stable high-KL operating point	Not reported	Documented; stabilization proposed	Not reported