RESEARCH APRIL 6, 2026

IMPROVING MULTI-TURN TOOL USE WITH GRPO: RESULTS AND INSIGHTS

By Razvan Dumitru, Chetan Rane, Sami Hassaan, Divyansh Agarwal

Executive Summary

We're sharing early insights from applying GRPO reinforcement learning to multi-turn tool-use tasks using our MCP Tool Use dataset.

In a controlled experiment with 3,000 samples, we fine-tuned Qwen2.5-14B using LoRA (rank 32) and evaluated it on MCP Atlas. We observed significant improvement in both coverage rate and pass rate. In this article, we share observations on how data quality, reward design, and training constraints interact in agentic training settings.

Key takeaways:

- High-quality, task-aligned datasets materially improve tool-use coverage and reliability.
- Rubric-based rewards with LLM judges provide an effective signal for multi-step reasoning.
- Practical training optimizations (e.g., output truncation, iteration limits, relaxed validation) significantly impact outcomes.

These insights translated into meaningful performance improvements on MCP Atlas:

- **Coverage rate:** 23.5% ’ 30.4% (+29%)
- **Pass rate (0.50):** 15.8% ’ 31.9% (+102%)
- **Pass rate (0.75):** 7.7% ’ 17.3% (+125%)

We view this as a step toward more systematic, eval-driven training for agentic models, and we will continue sharing learnings as we support more complex data types in our experimentation pipeline.

The Challenge: Multi Tool Use at Scale

Large language models have become proficient at simple function calling, but real-world agentic tasks demand more: complex, multi-step execution where the model must discover relevant tools, chain them together across dozens of steps, interpret intermediate results, and synthesize a final answer.

MCP Atlas evaluates exactly this capability. Each task presents the model with a user request and a set of MCP (Model Context Protocol) servers: databases, calendars, email clients, file systems, CRMs, and more. The model must figure out which tools to call, in what order, with what parameters, and when to stop. Success is measured by rubric-based criteria: did the model complete the essential steps required to solve the task?

The baseline Qwen2.5-14B model, despite being a capable general-purpose model, achieved only 23.5% coverage and 7.7% pass rate on MCP Atlas.

The MCP Tool Use Dataset

To close this gap, we built the MCP Tool Use training dataset, a collection of 3000 agent prompts and rubric verifiers designed specifically for training agentic capabilities. Here's what makes it distinctive:

- **Realistic trajectories**: Each task captures the full complexity of real agentic workflows, including tool discovery, error recovery, and multi-step reasoning.
- **Granular Rubric Annotations**: Every trajectory is annotated with numerous task-specific rubrics, providing both process and outcome rewards. This enables the dense reward signal that makes GRPO training effective.
- **Diverse Tool Environment**: The dataset covers a broad range of MCP server types, including CRM systems, calendar tools, email clients, weather APIs, and more. This diversity is what allows trained models to generalize across MCP Atlas's varied task categories rather than overfitting to a narrow set of tool patterns.

Our Approach: GRPO with Rubric Rewards

We used GRPO (Group Relative Policy Optimization) with rubric-based rewards.

Each trajectory (sample provided in the Appendix) was evaluated against task-specific rubrics containing essential and optional criteria. An LLM judge (Claude Sonnet 4.5) scored each rubric item as pass/fail based on the model's tool call sequence and outputs. The reward was the fraction of essential rubric items passed, providing a dense signal that captures partial progress toward task completion.

1. What the Model Learned:

Tool strategy became more targeted: The baseline model spread its tool calls across many tools with low success rates. After training, the model concentrated on high-value tools e.g. calculator for numerical tasks, targeted ID-based lookups instead of fuzzy searches, and domain-specific tools matched to the task type.

The model learned when to stop: Baseline trajectories frequently hit the iteration cap without calling the finish tool. After training, the majority of trajectories terminated with an explicit finish call, indicating the model had developed a sense of task completeness.

More deliberate tool usage: The trained model showed a higher ratio of successful-to-failed tool calls per trajectory, with fewer repeated calls to the same endpoint and more consistent use of correct parameters on the first attempt.

2. Results:

MCP Atlas Performance

The pass rate improved significantly at both thresholds, indicating that the model doesn't just attempt more rubric items, it completes them at a substantially higher quality level. Coverage rate, the share of rubric items

the model addresses at all, improved by 29%. This shows better average answering quality, while the larger pass rate gains show it is solving them better.

3. Key Insights:

Reward Signal Quality > Reward Signal Magnitude

Rubric-based LLM judge reward provided a much richer training signal than binary task completion. A trajectory that completes 3 out of 5 essential steps receives reward 0.6, not 0.0. This dense signal gives GRPO meaningful within-group variance for advantage estimation, even when no trajectory fully solves the task. Without this, the majority of training batches would provide zero gradient signal.

Context Management Is a First-Order Concern

Multi-turn agentic tasks consume context rapidly. Each tool call adds the request and response to the conversation history. Without truncation, a single verbose database query can consume half the available context. Managing context through output truncation, iteration limits, and prompt engineering is not an afterthought but a primary lever for training performance.

Outcome and process rubrics:

Process rewards (did you call the right tools in the right order?) and outcome rewards (did you get the final answer?) are entangled in the rubrics. A trajectory that correctly queries the database but fails to synthesize the final answer gets a partial reward and critically, the model learns which tool calls were relevant. Pure outcome reward would give zero signal on the 85%+ of trajectories that don't fully solve the task while pure process reward would incentivize going through the motions without caring about results.

Conclusion

Reinforcement learning with rubric-based rewards can substantially improve LLM performance on multi-turn tool-use tasks, more than doubling pass rates on MCP Atlas. The key ingredients are: high-quality reward signals, and pipeline optimizations that maximize the model's effective context for reasoning. The resulting model doesn't just call more tools; it learns which tools matter, how to use them efficiently, and when to stop.

Interested in our training dataset or future research direction? Reach out to labs@scale.com.

Appendix

Sample Task:

We’re doing a quick review of billing-related support issues and customer sentiment ahead of our mid-November business review.

Could you pull a few metrics for me, using 5:00 PM on November 12, 2025 as the cutoff period?

1.

Compare the number of tickets tagged “billing” which were created in October 2025 vs. the number created in November 2025 up until November 12, 2025 at 5:00PM. Calculate the percentage change between the two volumes.

2. There were some quality issues with agent responses in early October. Identify the earliest-resolved billing ticket in the above timeframe. For audit purposes, look at the first comment added to that ticket. When was the most recent date that the author of that comment logged into the platform?
3. From our customer history data, calculate the average feedback rate for tickets purchased during the October to November 12th period in 2025, and compare it to the same date range from the prior year.

Trajectory (14 total tool calls):

Step 0: Discovers the relevant tag with zendesk_get-tags() and confirms that billing is an available tag with 47 tickets.

Steps 1-3: Retrieves billing tickets using zendesk_get-tickets-by-tag(tag="billing"). The first page returns the billing ticket set, including ticket #46, created on 2025-10-14T02:20:06. A second-page request returns no additional tickets, and the model then repeats the first-page request.

Steps 4-6: Inspects ticket #46 more closely. It calls zendesk_get-ticket(ticket_id=46), then zendesk_get-ticket-comments(ticket_id=46) and identifies the first comment author as Aisha Malik. It then calls zendesk_get-user(user_id=5) and retrieves the user’s last login timestamp: 2025-10-14T17:36:06.

Step 7: Calls zendesk_get-ticket-stats() while attempting to gather ticket-level summary information.

Steps 8-12: Misroutes the customer-satisfaction portion of the task into the finance toolchain. The model searches for “VantageLens” twice with finance-engine_search_quotes, resolves the result to ticker IFF, retrieves ticker metadata with finance-engine_get_ticker_info(symbol="IFF"), and then requests monthly price history twice with finance-engine_get_price_history(symbol="IFF", period="1y", interval="1mo").

Step 13: Finishes with a final answer that correctly reports the first-comment author lookup path, but incorrectly states a 100% decrease in billing tickets and fails to retrieve the required feedback-rate metric.

Conclusion: The correct solution first retrieves the billing-tagged tickets and filters them to the requested time window, including identifying that there are 15 billing tickets created in November 2025 up to November 12, 2025 at 5:00 PM. It must then determine that ticket ID 48 was resolved earliest among the relevant billing tickets, trace the first comment on that ticket to the correct author, and identify that Mei Tan (user ID 7) has last login date 2025-10-15T06:49:06. For the customer-satisfaction portion, the rubric-backed answer key also requires identifying that customer 34 gave a feedback rate of 5. The sample trajectory therefore fails because it drills into ticket 46 instead of the correct ticket, follows the wrong first-comment author lookup path, misstates the billing-ticket comparison, and misroutes the feedback-rate subproblem into an unrelated finance toolchain.

Rubrics:

There are a total of 47 high-quality rubrics, we have omitted the rest for conciseness.