

Mock Worlds, Real Skills: Building Small Agentic Language Models with Synthetic Tasks, Simulated Environments, and Rubric-Based Rewards

Anonymous ACL submission

Abstract

Small LLMs often struggle to match the agentic capabilities of large, costly models. While reinforcement learning can help, progress has been limited by two structural bottlenecks: existing open-source agentic training data are narrow in task variety and easily solved; real-world APIs lack diversity and are unstable for large-scale reinforcement learning rollout processes. We address these challenges with SYNTHAGENT, a framework that jointly synthesizes diverse tool-use training data and simulates complete environments. Specifically, a strong teacher model creates novel tasks and tool ecosystems, then rewrites them into intentionally underspecified instructions. This compels agents to actively query users for missing details. When handling synthetic tasks, an LLM-based user simulator provides user-private information, while a mock tool system delivers stable tool responses. For rewards, task-level rubrics are constructed based on required subgoals, user-agent interactions, and forbidden behaviors. Across 14 challenging datasets in math, search, and tool use, models trained on our synthetic data achieve substantial gains, with small models outperforming larger baselines.¹

1 Introduction

Large language models (LLMs) demonstrate strong agentic capabilities within ReAct-style frameworks (Yao et al., 2023). Through an iterative *reasoning–action–observation* loop, LLM-based agents can solve complex tasks that require interaction with external environments (Xi et al., 2025), such as booking hotels or canceling flights (Barres et al., 2025). However, these agentic capabilities depend heavily on very large base models (Bai et al., 2025), resulting in substantial inference costs and deployment overhead. Consequently, enabling

¹Code for data synthesis pipeline and training: <https://anonymous.4open.science/r/SYNTHAGENT-68A4/>

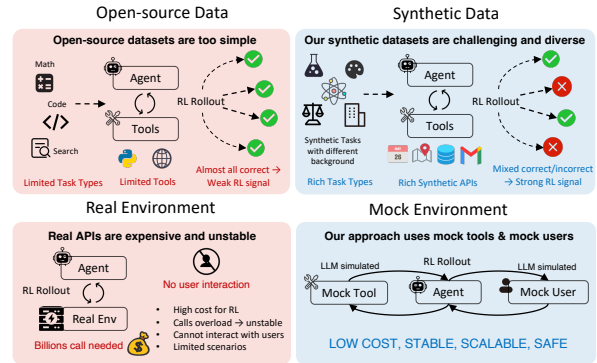


Figure 1: Comparison between existing agentic RL training recipes and ours. Open-source agentic training data are narrow in domain, while real-world APIs are costly and unstable. We replace these with diverse synthetic tasks and associated mock environments.

smaller models to reproduce the agentic capabilities of large models has become an important research direction (Lyu et al., 2025; Li et al., 2025).

Distillation methods based on supervised fine-tuning (SFT), in which a student model clones a teacher’s behavior (Torabi et al., 2018), can enhance the agentic capabilities of small models. Recent studies (Mai et al., 2025) further show that reinforcement learning (RL) is more effective than SFT for improving long-horizon planning and adaptive decision-making. However, most RL-based approaches focus on refining RL algorithms themselves (Dong et al., 2025a,c), while overlooking two fundamental bottlenecks:

- *Lack of diverse and challenging agentic training data.* Public datasets cover only a narrow range of domains and tools, and many have already been seen by modern LLMs during pre-training or fine-tuning. As a result, RL rollout often yields near-perfect trajectories with weak learning signals (Yu et al., 2025).
- *Absence of stable, diverse environments.* Real environments rarely support real-time model-

063	user interaction and offer only a narrow tool	horizon interactions, we train models with	113
064	set. RL rollout also requires a massive number	execution-grounded, rubric-based rewards.	114
065	of tool calls, making it impractical to rely on		
066	costly real-world APIs (LongCat, 2025).		
067	To address these, we introduce SYNTHAGENT,		
068	a framework that synthesizes tool-use tasks along		
069	with lightweight mock tool interfaces. A strong		
070	agentic teacher LLM generates novel tasks and		
071	their associated tools, guided by diverse persona		
072	backgrounds (Ge et al., 2025). As shown in Fig-		
073	ure 1, each task is paired with its own tool ecosys-		
074	tem, greatly expanding task and tool diversity.		
075	Moreover, the synthetic tools require no real de-		
076	ployment: an open-source LLM simulates both		
077	user and tool responses locally, ensuring stability.		
078	Specifically, for synthesizing training data, we		
079	introduce an information gap by rewriting detailed		
080	workflows as underspecified instructions, while		
081	critical details are hidden in a private user con-		
082	text. This design forces agents to actively query		
083	users and call tools to recover missing information,		
084	encouraging genuine long-horizon interaction. Sec-		
085	ond, for LLM-based tool response consistency, we		
086	maintain a task-level mapping of prior tool calls		
087	and responses. New calls are answered by consult-		
088	ing this mapping for consistent replies. As each		
089	synthetic task has a unique toolset, the mapping is		
090	scoped per task, keeping it lightweight during roll-		
091	out. Finally, for reward design, we avoid subjective		
092	LLM-written rubrics and derive rewards from ob-		
093	servable behavior. Using the workflow from data		
094	synthesis as a reference, we extract corresponding		
095	high-level subgoals from real execution trajectories,		
096	each reachable via multiple valid paths. This yields		
097	execution-grounded rewards that support diverse		
098	strategies, while filtering out low-quality data when		
099	the teacher fails to reliably complete the workflow.		
100	We evaluate our approach on 14 recent, chal-		
101	lenging datasets spanning agentic tool use (Yehu-		
102	dai et al., 2025) and short-horizon reasoning. In		
103	real-world tasks, models trained on synthetic data		
104	within virtual environments substantially outper-		
105	form those trained on open-source datasets. After		
106	training, our 8B–14B models surpass a 32B model		
107	on multiple agentic benchmarks. In summary, the		
108	major contributions of this work are as follows:		
109			
110	<ul style="list-style-type: none">• We introduce an open-source framework for		
111	<ul style="list-style-type: none">synthesizing diverse agentic tool-use tasks,		
112	<ul style="list-style-type: none">with stable, lightweight mock tool interfaces.		
		<ul style="list-style-type: none">• Extensive experiments on 14 challenging	115
		<ul style="list-style-type: none">datasets demonstrate that models trained	116
		<ul style="list-style-type: none">on synthetic data and virtual environments	117
		<ul style="list-style-type: none">achieve strong real-world performance.	118
		<h2>2 Related Work</h2>	119
		<h3>2.1 Agentic Reinforcement Learning</h3>	120
		Recent studies show that RL outperforms SFT	121
		in long-horizon planning and adaptive decision-	122
		making (Zhang et al., 2025), making RL a core	123
		paradigm for training LLM agents in dynamic,	124
		multi-turn environments (Mialon et al., 2023). Clas-	125
		sical methods such as Q-learning (Mnih et al.,	126
		2015), PPO (Schulman et al., 2017), and self-	127
		play (Silver et al., 2017) have provided the concep-	128
		tual foundation for agentic optimization in LLM-	129
		based systems. These techniques have evolved into	130
		language-centric RL frameworks, where natural-	131
		language reasoning steps, tool calls, and observa-	132
		tions are treated as latent states and actions (Yao	133
		et al., 2023; Zhang et al., 2025). Recent work	134
		has further improved RL algorithms to better	135
		couple exploration with robust tool use in long-	136
		horizon tasks, including verifiable-reward RL (Su	137
		et al., 2025), entropy-regularized policy optimiza-	138
		tion (Dong et al., 2025a), and agent-specific PPO/-	139
		GRPO variants (Dong et al., 2025c). Despite this	140
		progress, research remains largely focused on RL	141
		algorithms, with considerably less attention given	142
		to data and environment design.	143
		<h3>2.2 Synthetic Data for Agentic Training</h3>	144
		The effectiveness of agentic RL depends on high-	145
		quality, diverse data and environments, which re-	146
		main scarce (Yehudai et al., 2025). Early works	147
		such as Self-Instruct (Wang et al., 2023) use strong	148
		but closed-source LLMs to generate instruction-	149
		following data for training smaller open-source	150
		models. To further increase diversity, Ge et al.	151
		(2025) propose Persona Hub, which curates one	152
		billion web-derived personas to enable diverse syn-	153
		thetic data generation. In parallel, Qin et al. (2025)	154
		identify a “rectified scaling law” for synthetic data:	155
		as long as diversity is maintained, gains from syn-	156
		thetic pre-training persist even at very large scales.	157
		For environment construction, benchmarks (Qin	158
		et al., 2024) rely on real-world APIs for authen-	159
		ticity or use LLMs to simulate existing APIs and	160
		reduce cost. However, these environments remain	161

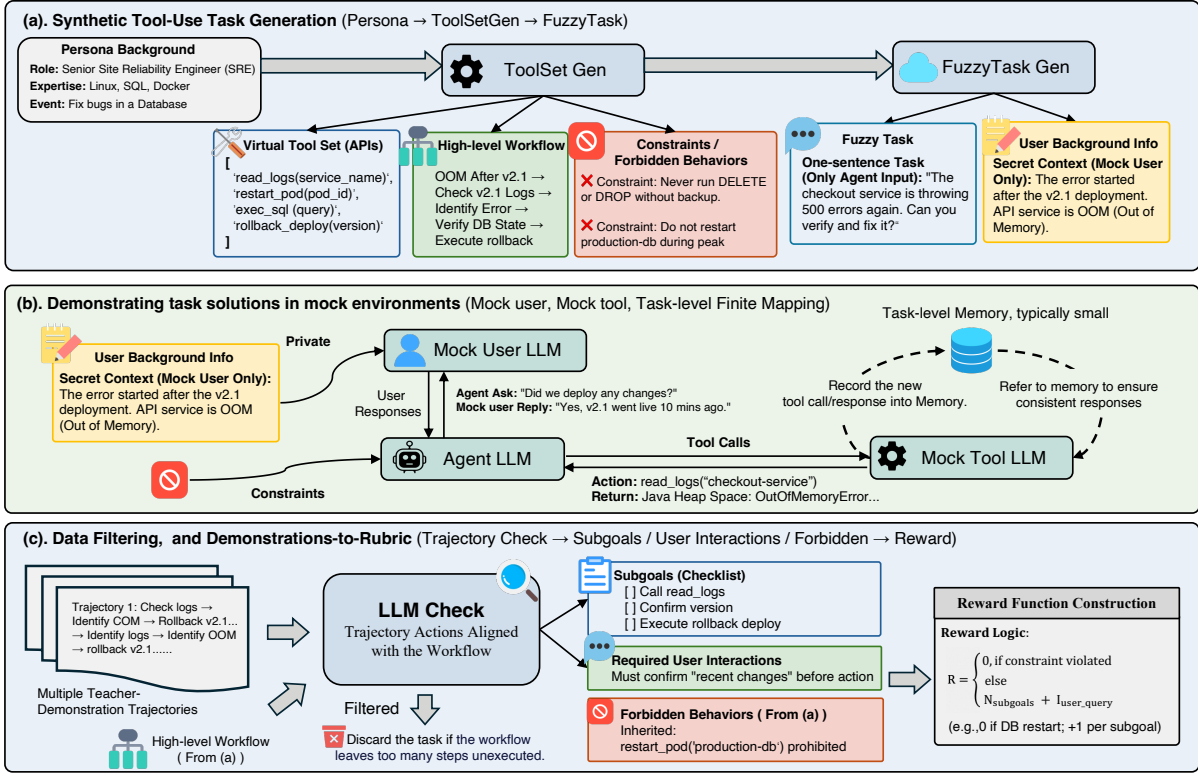


Figure 2: A unified pipeline for generating synthetic tool-use tasks, constructing stable mock environments, and deriving rubric-based rewards for agentic RL. Diverse tasks and tool ecosystems are created, guided by personas. For each synthetic task, an LLM-simulated user and environment are employed. To assign rewards, multiple trajectories are compared to the previously generated high-level workflow to infer task-specific rubrics.

constrained by real services, a lack of diversity (Lu et al., 2025). And identical states and actions can produce inconsistent responses in simulation, making them unsuitable for direct RL.

Most existing synthesis methods neither target long-horizon agentic tasks nor construct stable RL environments, meaning few works approach ours on either front. Our approach further unifies both aspects and integrates them with rubric-based RL.

3 Methodology

In this section, we move beyond improving RL algorithms themselves, and instead focus on two fundamental yet underexplored factors that limit agentic RL for LLMs: diverse, challenging training data and diverse, stable environments. Figure 2 illustrates SYNTHAGENT, a framework for synthesizing tool-use tasks, constructing virtual environments, and deriving rubric-based task rewards.

3.1 Synthetic Tool-Use Task Generation

Tool Set Generation Existing open-source agentic datasets are predominantly composed of web search and math tasks, which typically can be

solved using only a search tool or a code interpreter, resulting in homogeneous tool environments. To diversify settings and tool usage, we incorporate large-scale personas from Persona Hub (Ge et al., 2025) as backgrounds (Figure 2(a)). These personas cover a wide range of identities and scenarios (e.g., a senior SRE debugging a database issue).

For each persona-defined context, we employ a strong agentic LLM to (i) infer a high-level workflow describing how an individual with that background would accomplish the task, and (ii) based on this inferred workflow, construct a task-specific virtual tool ecosystem with tool descriptions and API specifications. As a result, each task is paired with a dedicated tool suite, encouraging models to learn tool-use procedures rather than memorize fixed APIs. To further increase task difficulty, we introduce task-level forbidden constraints, such as “disallowing a system reboot during a database repair task”. These constraints raise the RL challenge by requiring the model to plan and act within non-trivial restrictions.²

²For data synthesis, we employ Qwen3-235B-A22B-Instruct-2507 (Yang et al., 2025) due to its strong agentic capabilities and low-cost local deployment.

Fuzzy Task Generation After defining the tool set available to the agent, we next design the tasks, i.e., the agent’s inputs. Tasks derived directly from previously generated high-level workflows are often over-specified; for example, *Check logs* \rightarrow *Verify DB state* \rightarrow *Execute rollback*. The initial input outlines the workflow, and simply following it becomes the optimal action sequence a^* . Consequently, rollouts $\tau \sim \pi_\theta$ are highly homogeneous. At many steps t , the policy $\pi_\theta(a | s_t)$ is nearly deterministic; thus, during RL training, the variation of the advantage under π_θ may become negligible:

$$\text{Var}_{a_t \sim \pi_\theta(\cdot | s_t)}[A(s_t, a_t)] \approx 0, \quad (1)$$

where $A(s_t, a_t)$ is the advantage under a value baseline, satisfying $\mathbb{E}_{a_t \sim \pi_\theta(\cdot | s_t)}[A(s_t, a_t)] = 0$. As demonstrated by the Cauchy–Schwarz inequality:

$$\begin{aligned} & \left\| \mathbb{E}_{a_t \sim \pi_\theta(\cdot | s_t)}[\nabla_\theta \log \pi_\theta(a_t | s_t) A(s_t, a_t)] \right\| \\ & \leq \sqrt{\mathbb{E}_{a_t \sim \pi_\theta(\cdot | s_t)}[\|\nabla_\theta \log \pi_\theta(a_t | s_t)\|^2]} \\ & \quad \cdot \sqrt{\mathbb{E}_{a_t \sim \pi_\theta(\cdot | s_t)}[\|A(s_t, a_t)\|^2]} \\ & = \sqrt{\mathbb{E}_{a_t \sim \pi_\theta(\cdot | s_t)}[\|\nabla_\theta \log \pi_\theta(a_t | s_t)\|^2]} \\ & \quad \cdot \sqrt{\text{Var}_{a_t \sim \pi_\theta(\cdot | s_t)}[A(s_t, a_t)]}, \end{aligned} \quad (2)$$

when Eq. (1) holds, the expected gradient magnitude shrinks, weakening the learning signal.

To mitigate this degeneracy, we inject an *information gap* during task construction: partition each task initial state s_0 into an agent-visible instruction I and a user-only hidden context C :

$$s_0 \mapsto (I, C), \quad \text{s.t. } H(a^* | I) \gg \epsilon. \quad (3)$$

Initially, I is insufficient to determine the optimal action a^* uniquely; the conditional entropy $H(a^* | I)$ is large. Critical details must be recovered through interaction. As illustrated in Figure 2(a), we employ an LLM to rewrite an overly explicit request into a minimal one I , for example, “The checkout service is returning 500 errors again. Can you investigate and fix it?” Decisive details (e.g., “v2.1 was just deployed” and “OOM”) are moved to C and revealed only when the agent queries the user.

Under intentionally high $H(a^* | I)$, the policy must first query for the missing context C before invoking tools. As observations $o_{\leq t}$ gradually reveal the hidden information, uncertainty decreases ($H(a^* | I, o_{\leq t}) < H(a^* | I)$). Early decisions

therefore become nontrivial: the agent must decide which clarification to ask first to elicit informative observations o_t , preventing $\pi_\theta(\cdot | s_t)$ from becoming near-deterministic. This mitigates the gradient degeneration issue during model training.

Appendix C provides examples of the synthetic tasks and tools.

3.2 Mock Environments

Mock Tool & User When the model attempts the synthetic tool-use tasks described above, the corresponding tool set is registered in the system prompt. Because these tools are virtual rather than real, we must simulate both tool execution and responses. To this end, we build a fully LLM-simulated mock environment, which requires no real deployment and supports large-scale interaction during RL rollout. The LLM-simulated tool receives the model’s tool-call requests and returns appropriate outputs. The LLM-simulated user answers the model’s queries based on user-private background information C generated earlier.³

Stable Environments During the RL rollout process, the same task is attempted many times, raising a central concern: if tool responses are non-reproducible, then even under the same state s , executing the same tool action a (identical `tool + args`) may yield different observations o . This randomness propagates along the trajectory, so that the same (s_t, a_t) can induce different future returns across rollouts, making the advantage estimate \hat{A}_t inconsistent. Consequently, even for identical (s_t, a_t) , the policy update term $\hat{g}_t = \nabla_\theta \log \pi_\theta(a_t | s_t) \hat{A}_t$ may exhibit substantially different magnitudes and even opposite signs across samples, hindering training stability.

A natural mitigation is to add retrieval-augmented memory (Lewis et al., 2020) to the tool simulator, storing past tool calls and responses. When generating a new tool response, the simulator retrieves similar calls as references to ensure consistency. In our setting, each task has its own tool suite, so only a few calls require within-task consistency. Instead of a full memory system, we use a lightweight task-level finite mapping:

$$\mathcal{M} = \{(u_i, y_i)\}_{i=1}^M, \quad u_i = (\text{tool}_i, \text{args}_i). \quad (4)$$

³These interactions are primarily simple, formatted question-answering tasks requiring no powerful model. Thus, we implement this component using Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025), which is easy to deploy locally and incurs very low runtime cost.

When the model issues a valid tool call u , the simulator checks similar entries in \mathcal{M} and checks for a semantically equivalent one. If none is found, it generates a response y and adds (u, y) to the mapping. Equivalence checking and response generation can be handled in a single forward pass, adding no extra computational cost.

Empirically, the task-level mapping remains very small. For example, in a rollout with 16 trajectories and an average of 10 tool calls per trajectory, even if all calls were unique, the size of \mathcal{M} would be at most 160. Therefore, we can *omit retrieval* altogether and include \mathcal{M} directly in the tool simulator’s prompt, allowing the model to identify matches. The entire process remains lightweight and efficient, significantly improves training stability without restricting exploration.

3.3 Automatic Rubric-Based Rewards

Task-Level Rubrics Unlike math tasks with clearly defined correctness-based rewards (Guo et al., 2025), reward design for multi-step tool-use tasks is inherently challenging. A common practice is to use an LLM as a judge to assign a scalar score to each trajectory, but such judgments can be subjective. Fortunately, we construct fuzzy tasks by rewriting high-level workflows, whose steps provide objective subgoals (e.g., *Check logs*, *Verify DB state*, *Execute rollback*) for trajectory rewards.

However, the designed workflow may not match real executions. To address this, we collect multiple actual executed trajectories from strong teacher models. Using the workflow as a reference, we prompt an LLM to extract workflow-relevant subgoals and user-agent interactions from each trajectory. Both are grounded in the workflow and can be achieved via multiple exploration paths.

If the workflow leaves too many steps unexecuted, the example is removed. This filters noisy data and reduces reliance on the number of teacher demonstrations, since trajectories with many unexecuted steps can be discarded.

Additionally, during tool set generation (Section 3.1), each task is paired with its own set of forbidden behaviors (e.g., disallowing a system reboot). Combined with the subgoals and interaction requirements, these form a complete task-level rubric unique to each synthetic task. Appendix C provides examples of the generated rubrics.

Rubric-based Reward During RL training, we use an LLM as a judge to assign rewards based on

the task-level rubric. Specifically,

$$R(\tau) = \mathbb{I}(\tau) \cdot (N_{\text{subgoals}}(\tau) + I_{\text{user_query}}(\tau)). \quad (5)$$

Here, $\mathbb{I}(\tau) \in \{0, 1\}$ is 0 if and only if τ violates any rubric-specified forbidden behavior (yielding zero reward), and 1 otherwise. $N_{\text{subgoals}}(\tau) \in [0, 1]$ is the fraction of subgoals completed, and $I_{\text{user_query}}(\tau) \in [0, 1]$ is the fraction of required user-agent interactions satisfied; we average these scores as the final reward. This rubric-based reward design scales seamlessly to large numbers of synthesized tasks.

Table 1 summarizes the statistics of our synthesized tool-use dataset, and reports the total token cost of the full synthesis pipeline (including tool, task, and rubric generation). Since the entire process runs on locally deployed open-source models, the cost is negligible. And even with commercial APIs, it remains well within an affordable range.

3.4 Final RL Training

Following the technical reports of Kimi K2 (Bai et al., 2025) and LongCat (LongCat, 2025), strong reasoning ability is essential for agentic tasks. Accordingly, we augment our virtual tool-use data with a small set of high-difficulty reasoning tasks, sampling 4,000 search or math instances from ToolStar (Dong et al., 2025b). Since our tool-use data contain rich contextual backgrounds whereas math problems are purely abstract, this mismatch may hinder training. To increase diversity, we prompt Qwen3-235B⁴ with persona information (Ge et al., 2025) to rewrite each problem into a scenario-based variant. Each synthesized problem is then solved 3 times by Qwen3-235B, and we retain only those with fully consistent answers, ensuring reliability.

Tool-use tasks run in a virtual environment with rubric-based rewards, while reasoning tasks run in a real Python environment and are evaluated by answer correctness. We then train the model on the combined dataset using GRPO (Shao et al., 2024).

4 Experiments

4.1 Experimental Setup

We evaluate SYNTHAGENT by assessing models trained on our synthetic data and within simulated environments. Our experiments focus on agentic benchmarks that measure long-horizon tool use, multi-turn planning, and adaptability to unfamiliar

⁴For brevity, we use Qwen3-235B to represent Qwen3-235B-A22B-Instruct. The same applies in the following text.

Statistic	Value
Total #Tasks	15,096
Avg. #Tools Per Task	4.1
Avg. #Interactions (After Training)	13.4
Avg. Mapping Size $ M $ (After Training)	30.1
Avg. Per-Task Token Usage in Synthesis	25,278

Table 1: Statistics of the synthesized tool-use dataset.

tools. We also test short-horizon generalization through reasoning tasks such as math and search.

Agentic Tool Use Benchmarks We evaluate on the most representative agentic benchmarks: TAU-2 (Barres et al., 2025) and BFCL-V4 (Patil et al., 2025), spanning 7 datasets and nearly 100 diverse tools. These benchmarks are widely used by Qwen (Yang et al., 2025), Kimi (Bai et al., 2025), and DeepSeek (DeepSeek-AI, 2025), aligning our protocol with that of leading foundation models.

BFCL-V4 provides multiple datasets; we focus on its multi-turn subset (about 800 tasks). These tasks span diverse real-world domains such as trading, vehicle control, and social media. Each task typically requires 5 to 20 tool-interaction turns, providing a rigorous evaluation of the model’s capabilities in parameter clarification and error rejection.

TAU-2 targets three real-world business domains: airline, retail, and telecommunications, comprising roughly 300 tasks. These tasks generally require multi-turn interactions between the agent and user. Moreover, users can also invoke tools and modify the environment, meaning the model must not only execute tools correctly but also guide the user and handle uncertain feedback.

The above agentic benchmarks, with their unfamiliar tools and long-horizon planning demands, serve as our primary evaluation suite.

Reasoning Benchmarks We also examine the short-horizon reasoning capabilities of our framework. We employ several math benchmarks (AIME24, AIME25, HMMT25, Olympiad (He et al., 2024)) and search benchmarks (FRAMES (Krishna et al., 2025), WebWalker (Wu et al., 2025), Xbench (Chen et al., 2025b)). These tasks involve only two tools: a Python interpreter and Google Search and typically require fewer than five interaction turns. However, each step demands deeper reasoning than in the agentic benchmarks, making them suitable for out-of-domain evaluation.

All benchmarks were introduced after 2024, ensuring strong relevance and up-to-date evaluation.

Evaluation TAU-2 and BFCL provide not only datasets but also full interactive environments. During evaluation, the model must invoke tools to interact with these environments; performance is measured by checking whether environment states are correctly updated to their ground-truth values using Exact Match. For math reasoning, we also apply Exact Match. For more free-form outputs in search reasoning, we use Qwen3-235B to judge whether the model’s responses are semantically consistent with the ground truth. The search tool is implemented via the Google Search API.

Baselines We compare our model, trained on synthetic tasks and mock environments, against the following baselines: RL-trained models on the latest open-source ToolStar (Dong et al., 2025b), which employ 30,000 math and search examples. We also evaluate strong LLMs prompted to integrate tools, such as the larger Qwen3-32B. The Qwen3 technical report (Yang et al., 2025) indicates that Qwen3-32B has already been trained on synthetic tool-use data using RL, making it a competitive baseline. All baselines perform inference using the OpenAI function-calling format and the same prompt, ensuring a fully consistent setup.

Implementation Using SYNTHAGENT, we generate 15,096 synthetic tool-use tasks entirely with locally deployed open-source LLMs. We train Qwen3-8B/14B with GRPO (non-thinking) to assess data quality. For rubric construction, we collect four demonstrations from a strong agentic teacher (Qwen3-235B). Since rubric design depends mainly on high-level workflow rather than specific teacher demonstrations, the number of them typically has little impact, as shown in Section 4.3. For reward, we employ Qwen3-30B-A3B-Instruct to judge with rubrics. More implementation details are provided in Appendix A.

4.2 Main Results

Table 2 reports the performance of our 8B and 14B models, trained on synthetic data and simulated environments, on real-world agentic benchmarks such as TAU-2 and BFCL-Multi-turn.

SYNTHAGENT enables small models to match and even surpass much larger agentic models. Using synthetic tool-use tasks and fully simulated environments, our method yields substantial gains

Method	TAU-2 Bench			BFCL-V4-Multi-turn				Avg.
	Airline	Telecom	Retail	Base	Miss Func	Miss Param	Long Context	
Baselines trained on closed-source or the latest open-source data (non-thinking, using tools)								
Qwen3-235B	47.5	37.7	68.0	58.5	47.5	35.0	54.0	49.7
Qwen3-14B	22.0	25.4	39.5	40.0	34.5	26.5	26.5	30.6
Qwen3-32B	22.5	27.6	44.7	50.5	43.0	30.5	33.0	36.0
ToolStar-8B	13.5	25.9	39.5	52.0	38.0	22.5	30.5	31.7
ToolStar-14B	18.0	30.7	40.4	56.5	35.5	29.5	39.5	35.7
SYNTHAGENT-8B	34.5	38.2	57.2	54.5	45.5	33.0	37.5	42.9
SYNTHAGENT-14B	40.0	44.7	58.6	57.0	46.5	31.0	46.0	46.3

Table 2: Agentic performance comparison. For TAU-2 (Airline, Telecom, Retail), we report Avg@4 and use the open-source Kimi-K2-Instruct model as the user simulator. Qwen3-235B refer to the model Qwen3-235B-A22B-Instruct. The same applies in the following text. Best results except Qwen3-235B are **bolded**.

Method	Math				Search			Avg.
	AIME24	AIME25	Olympiad	HMMT25	Frames	XBench	WebWalker	
Baselines trained on closed-source or the latest open-source data (non-thinking, using tools)								
Qwen3-235B	83.3	70.6	83.4	64.4	70.5	43.0	59.5	67.8
Qwen3-14B	39.4	35.6	53.8	39.4	37.8	21.0	38.0	37.9
Qwen3-32B	50.0	41.1	56.4	35.1	44.8	25.0	37.0	41.3
ToolStar-8B	60.6	54.4	76.4	47.8	58.5	33.0	44.5	53.6
ToolStar-14B	71.7	63.3	77.3	45.0	60.4	40.0	44.0	57.4
SYNTHAGENT-8B	71.6	58.9	77.2	48.9	59.7	45.0	48.5	58.5
SYNTHAGENT-14B	72.2	66.7	80.1	53.9	63.5	43.0	50.0	61.3

Table 3: Short-horizon reasoning performance comparison (math and search). For AIME24, AIME25, and HMMT25, we report Avg@6 for more stable evaluation. Best results except Qwen3-235B are **bolded**.

477 across real-world agentic benchmarks. On TAU-2
478 and BFCL-Multi-turn, SYNTHAGENT-8B scores
479 42.9 on average (+12.3 over Qwen3-14B, and +6.9
480 over Qwen3-32B). The improvements are even
481 greater with SYNTHAGENT-14B, which scores
482 46.3 on average. Despite being far smaller, SYN-
483 THAGENT-14B matches Qwen3-235B on many
484 TAU-2 and BFCL domains, demonstrating that our
485 synthetic training strategy can close the gap with
486 much larger models.

487 **Synthetic data and simulated environments**
488 **substantially outperform existing open-source**
489 **datasets.** Open-source agentic datasets cover only
490 a limited set of tools and cannot capture the com-
491 plexity of real tool-use logic. For example, in the
492 latest ToolStar (Dong et al., 2025b) dataset, RL-
493 trained models show no significant improvement
494 in agentic performance. In contrast, SYNTHA-
495 GENT-8B achieves great performance in TAU-2,

496 consistently surpassing open-source baselines, with
497 BFCL-Multi-turn showing the same pattern. These
498 results indicate that diverse and challenging syn-
499 thetic tasks are far more effective for strengthening
500 a model’s agentic abilities.

501 **Beyond tool-use tasks, the model also general-**
502 **izes to short-horizon reasoning.** Table 3 shows
503 that, although math and search tasks are not the
504 primary focus of our training(only 4,000 instances
505 sampled from ToolStar are included), SYNTHA-
506 GENT still achieves substantial gains. Under the
507 same non-thinking setting, SYNTHAGENT-8B sig-
508 nificantly outperforms Qwen3-14B and even ex-
509 ceeds an 8B model trained on 30,000 ToolStar ex-
510 amples. The improvements for SYNTHAGENT-
511 14B are even larger. These results indicate that
512 our method transfers effectively to new reasoning
513 domains, with tool-use data also exhibiting strong
514 generalization in reasoning tasks.

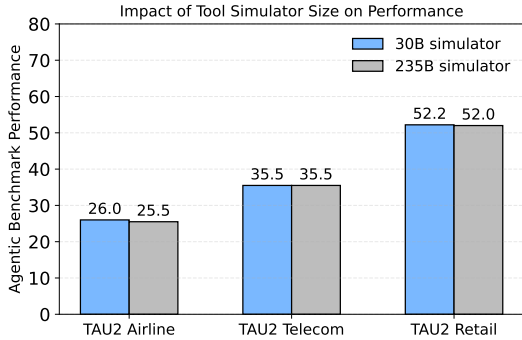


Figure 3: Effect of tool-simulator size on TAU-2 performance(5,000 training samples), showing negligible gains from larger simulators.

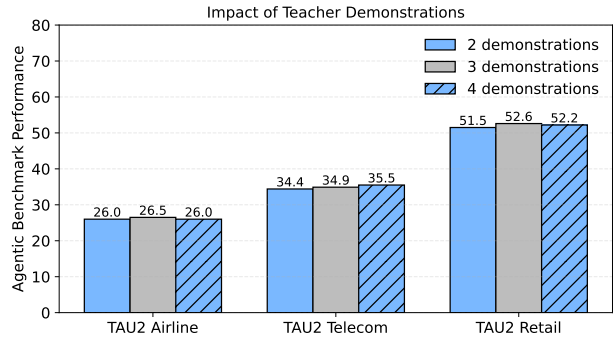


Figure 4: Influence of teacher-demonstration count when constructing task-level rubrics(5,000 training samples), indicating that additional demonstrations yield limited gain.

Method	TAU-2 Bench			BFCL-V4 Multi-turn Avg.
	Airline	Telecom	Retail	
Qwen3-8B	15.0	8.8	32.5	28.1
reasoning data only	22.0	26.1	38.2	37.9
W/O information gap	22.0	28.9	35.3	39.1
SYNTHAGENT-8B	34.5	38.2	57.2	42.6

Table 4: Ablation study on agentic benchmarks.

Overall, training on synthetic tasks and environments allows 8B–14B models to rival or surpass 32B models, drastically reducing inference cost.

4.3 Further Analysis

Ablation The results in Table 4 confirm that synthetic tool-use data is crucial. Using only reasoning data yields improvements over the Qwen3-8B baseline, mainly from short-term reasoning, but it remains inadequate for multi-turn long-horizon agentic tasks. In contrast, adding synthetic tool-use data provides substantial gains and consistently improves performance across all agentic benchmarks. However, if we do not introduce information gaps, do not rewrite workflows into less explicit descriptions, and directly use them for training without user interaction, the benefit of tool-use data becomes negligible. This further validates the rationale behind our design.

Impact of Tool Simulator Size on Performance

The tool simulator primarily generates responses to new tool calls, and checks whether a call matches a previous query in the prompt. Both are simple, well-defined operations that typically do not require a strong model. To validate this, we evaluate Qwen3-8B trained on 5,000 synthetic agentic tasks with two simulators: Qwen3-235B and the smaller

Qwen3-30B-A3B-Instruct model. As shown in Figure 3, a larger simulator does not improve performance, suggesting mock tool simulation is largely formatted QA and semantic matching, rather than a capability that benefits from model scale.

Impact of Number of Teacher Demonstrations

The rubrics and subgoals are mainly derived from the high-level workflow in Section 3.1. Teacher trajectories are used only to align the synthesized workflow with real executions; examples where the teacher covers too little of the workflow are discarded. In principle, rubric quality depends weakly on the number of teacher trajectories.

To evaluate this, we build rubrics for 5,000 synthetic agentic examples using 2, 3, or 4 teacher trajectories and compare performance. As shown in Figure 4, adding more demonstrations yields no significant gains, suggesting rubric construction does not require substantial computation.

More experiments, such as data-scaling effects on RL and RL-SFT comparisons at equal data sizes, are provided in Appendix B.

5 Conclusion

We present SYNTHAGENT, a novel framework addressing two core bottlenecks in training agentic language models: the scarcity of diverse, challenging tasks and stable tool environments. By jointly synthesizing tool-use tasks with underspecified instructions and providing stable mock environments, SYNTHAGENT enables efficient reinforcement learning for small models. Extensive evaluations demonstrate that models trained entirely on synthetic data and virtual environments achieve substantial gains, with small models surpassing much larger baselines.

576 Limitations

577 Agentic training data synthesis is an increasingly
578 important research topic. Technical reports from
579 leading foundation models (Qwen3 (Yang et al.,
580 2025), LongCat (LongCat, 2025), Kimi K2 (Bai
581 et al., 2025), DeepSeek V3.2 (DeepSeek-AI, 2025),
582 Minimax M2 (Chen et al., 2025a)) consistently
583 show that synthetic data, rather than real-world
584 data, forms the core of agentic RL. However, these
585 models do not release their synthetic datasets, nor
586 do they provide detailed descriptions of their syn-
587 thesis procedures. This limitation restricts our abil-
588 ity to refine the SYNTHAGENT pipeline based on
589 prior work and makes it difficult to compare against
590 stronger baselines. Future work should explore ad-
591 ditional approaches for agentic training data syn-
592 thesis and identify the key factors that are most
593 critical for building effective agents.

594 References

595 Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen,
596 Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun
597 Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao
598 Ding, Mengnan Dong, Angang Du, Chenzhuang Du,
599 Dikang Du, Yulun Du, Yu Fan, Yichen Feng, and 80
600 others. 2025. Kimi k2: Open agentic intelligence.
601 *arXiv preprint arXiv:2507.20534*.

602 Victor Barres, Honghua Dong, Soham Ray, Xujie Si,
603 and Karthik Narasimhan. 2025. τ^2 -bench: Evaluat-
604 ing conversational agents in a dual-control environ-
605 ment. *arXiv preprint arXiv:2506.07982*.

606 Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang,
607 Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao
608 Wang, Cheng Zhu, Chengjun Xiao, Chengyu Du,
609 Chi Zhang, Chu Qiao, Chunhao Zhang, Chunhui
610 Du, Congchao Guo, Da Chen, Deming Ding, and
611 80 others. 2025a. Minimax-m1: Scaling test-time
612 compute efficiently with lightning attention. *arXiv*
613 *preprint arXiv:2506.13585*.

614 Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Hao-
615 tong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang,
616 Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou,
617 Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng
618 Liu, Jinpeng Li, Kangyi Liu, and 14 others. 2025b.
619 xbench: Tracking agents productivity scaling with
620 profession-aligned real-world evaluations. *arXiv*
621 *preprint arXiv:2506.13651*.

622 DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the fron-
623 tier of open large language models. *arXiv preprint*
624 *arXiv:2512.02556*.

625 Guanting Dong, Licheng Bao, Zhongyuan Wang,
626 Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang,
627 Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui

Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 628
2025a. Agentic entropy-balanced policy optimiza- 629
tion. *arXiv preprint arXiv:2510.14545*. 630

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, 631
Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui 632
Zhou, Zhicheng Dou, and Ji-Rong Wen. 2025b. 633
Tool-star: Empowering llm-brained multi-tool rea- 634
soner via reinforcement learning. *arXiv preprint* 635
arXiv:2505.16410. 636

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, 637
Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Ji- 638
azhen Du, Huiyang Wang, Fuzheng Zhang, Guorui 639
Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 640
2025c. Agentic reinforced policy optimization. 641
arXiv preprint arXiv:2507.19849. 642

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao 643
Mi, and Dong Yu. 2025. Scaling synthetic data cre- 644
ation with 1,000,000,000 personas. *arXiv preprint* 645
arXiv:2406.20094. 646

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, 647
Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, 648
Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, 649
Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhu- 650
oshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. 651
Deepseek-r1 incentivizes reasoning in llms through 652
reinforcement learning. *Nature*, 645(8081):633–638. 653

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, 654
Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie 655
Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan 656
Liu, and Maosong Sun. 2024. OlympiadBench: 657
A challenging benchmark for promoting AGI with 658
olympiad-level bilingual multimodal scientific prob- 659
lems. In *Proceedings of the 62nd Annual Meeting of*
660 *the Association for Computational Linguistics (Vol-*
661 *ume 1: Long Papers)*, pages 3828–3850. 662

Satyapriya Krishna, Kalpesh Krishna, Anhad Mo- 663
hananey, Steven Schwarcz, Adam Stambler, Shyam 664
Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, 665
and reason: A unified evaluation of retrieval- 666
augmented generation. In *Proceedings of the 2025*
667 *Conference of the Nations of the Americas Chap-*
668 *ter of the Association for Computational Linguistics:*
669 *Human Language Technologies (Volume 1: Long Pa-*
670 *pers)*. 671

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio 672
Petroni, Vladimir Karpukhin, Naman Goyal, Hein- 673
rich Küttler, Mike Lewis, Wen-tau Yih, Tim Rock- 674
täschel, Sebastian Riedel, and Douwe Kiela. 2020. 675
Retrieval-augmented generation for knowledge- 676
intensive nlp tasks. *Advances in neural information*
677 *processing systems*, 33:9459–9474. 678

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, 679
Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qian- 680
ben Chen, Weichen Sun, Qiexiang Wang, Hongxuan 681
Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying 682
Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, 683
and 11 others. 2025. Chain-of-agents: End-to-end 684

685	agent foundation models via multi-agent distillation and agentic rl. <i>arXiv preprint arXiv:2508.13167</i> .	Scaling laws of synthetic data for language models. <i>arXiv preprint arXiv:2503.19551</i> .	741
686			742
687	Team Meituan LongCat. 2025. Longcat-flash technical report. <i>arXiv preprint arXiv:2509.01322</i> .	John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. <i>arXiv preprint arXiv:1707.06347</i> .	743
688			744
689	Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. 2025. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. In <i>Findings of the Association for Computational Linguistics: NAACL 2025</i> , pages 1160–1183.	Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. <i>arXiv preprint arXiv:2402.03300</i> .	745
690			746
691			747
692			748
693			749
694			750
695			751
696			
697	Yuanjie Lyu, Chengyu Wang, Jun Huang, and Tong Xu. 2025. From correction to mastery: Reinforced distillation of large language model agents. <i>arXiv preprint arXiv:2509.14257</i> .	Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. <i>arXiv preprint arXiv:2409.19256</i> .	752
698			753
699			754
700			755
701			756
702	Xinji Mai, Haotian Xu, Zhong-Zhi Li, Xing W, Weinong Wang, Jian Hu, Yingying Zhang, and Wenqiang Zhang. 2025. Agent rl scaling law: Agent rl with spontaneous code execution for mathematical problem solving. <i>arXiv preprint arXiv:2505.07773</i> .	David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. <i>arXiv preprint arXiv:1712.01815</i> .	757
703			758
704			759
705			760
706	Grégoire Mialon, Clémentine Fourier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In <i>The Twelfth International Conference on Learning Representations</i> .	Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. 2025. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. <i>arXiv preprint arXiv:2503.23829</i> .	761
707			762
708			763
709			764
710			765
711	Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dhharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. <i>Nature</i> , 518(7540):529–533.	Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral cloning from observation. In <i>Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence</i> , pages 4950–4957. International Joint Conferences on Artificial Intelligence Organization.	766
712			767
713			768
714			769
715			770
716			771
717			772
718			773
719			774
720	OpenAI. 2023. Gpt-4 technical report. <i>arXiv preprint arXiv:2303.08774</i> .	Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khachatryan, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In <i>Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)</i> , pages 13484–13508.	775
721			776
722	Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, 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 <i>Forty-second International Conference on Machine Learning</i> .	Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. Web-walker: Benchmarking llms in web traversal. In <i>Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , ACL 2025, Vienna, Austria, July 27 - August 1, 2025.	777
723			778
724			779
725			780
726			781
727			
728	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 <i>The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024</i> .	Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, and 9 others. 2025. The rise and potential of large language model based agents: A survey. <i>Science China Information Sciences</i> , 68(2):121101.	782
729			783
730			784
731			785
732			786
733			787
734			788
735			789
736			
737	Zeyu Qin, Qingxiu Dong, Xingxing Zhang, Li Dong, Xiaolong Huang, Ziyi Yang, Mahmoud Khademi, Dongdong Zhang, Hany Hassan Awadalla, Yi R. Fung, Weizhu Chen, Minhao Cheng, and Furu Wei. 2025.		790
738			791
739			792
740			793

798	An Yang, Anfeng Li, Baosong Yang, Beichen Zhang,	Hub (Ge et al., 2025). During toolset genera-	850
799	Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,	tion, we define virtual tools following the Ope-	851
800	Chengen Huang, Chenxu Lv, Chujie Zheng, Dayi-	nAI function-calling specification and filter out any	852
801	heng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge,	samples that cannot be parsed into a valid function-	853
802	Haoran Wei, Huan Lin, Jialong Tang, and 40 oth-	calling format. In total, we generated 15,096 tool-	854
803	ers. 2025. Qwen3 technical report. <i>arXiv preprint</i>	use tasks, along with a smaller set of math or search	855
804	<i>arXiv:2505.09388</i> .	reasoning tasks (approximately 4,000). All data	856
805	Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak	were produced locally with open-source LLMs,	857
806	Shafran, Karthik Narasimhan, and Yuan Cao. 2023.	without reliance on commercial APIs, ensuring a	858
807	React: Synergizing reasoning and acting in language	stable and cost-efficient pipeline.	859
808	models. In <i>International Conference on Learning</i>		
809	<i>Representations (ICLR)</i> .	Benchmarks Detailed benchmark statistics (e.g.,	860
810	Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun	subsets and test sizes) are reported in Table 5.	861
811	Zhao, Roy Bar-Haim, Arman Cohan, and Michal	Agentic tool-use benchmarks (TAU-2 and BFCL-	862
812	Shmueli-Scheuer. 2025. Survey on evaluation of llm-	V4 Multi-turn) measure long-horizon, multi-turn	863
813	based agents. <i>arXiv preprint arXiv:2503.16416</i> .	interaction with unfamiliar tools. Reasoning bench-	864
814	Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan,	marks (math and search) evaluate out-of-domain	865
815	Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan,	generalization, where tasks typically involve fewer	866
816	Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin,	tool calls but require deeper multi-step reasoning	867
817	Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan	and evidence synthesis.	868
818	Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and		
819	16 others. 2025. Dapo: An open-source llm rein-	Training We perform reinforcement learning us-	869
820	forcement learning system at scale. <i>arXiv preprint</i>	ing the GRPO algorithm (Shao et al., 2024) within	870
821	<i>arXiv:2503.14476</i> .	the VERL framework (Sheng et al., 2024). We use	871
822	Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin,	a global batch size of 128, a PPO mini-batch size of	872
823	Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li,	16, a rollout size of 16, and a maximum response	873
824	Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen,	length of 13,000 tokens, training for 2 epochs on	874
825	Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang,	8×NVIDIA H20 GPUs. The clipping ratio is con-	875
826	Yue Liao, Hongru Wang, Mengyue Yang, and 6	strained between 0.2 and 0.28, and we allow up	876
827	others. 2025. The landscape of agentic reinforce-	to 16 turns per rollout. Since Qwen3 base models	877
828	ment learning for llms: A survey. <i>arXiv preprint</i>	already exhibit strong tool/function-calling capa-	878
829	<i>arXiv:2509.02547</i> .	bilities and reliably follow the required format, we	879
830	Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue	skip the SFT phase for format learning and directly	880
831	Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos	proceed with RL training.	881
832	Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark W.		
833	Barrett, and Ying Sheng. 2024. Sglang: Efficient	Inference During inference, the model was de-	882
834	execution of structured language model programs. In	ployed with SGLang (Zheng et al., 2024) to in-	883
835	<i>Advances in Neural Information Processing Systems</i>	crease throughput. TAU-2 (Barres et al., 2025) and	884
836	<i>38: Annual Conference on Neural Information Pro-</i>	BFCL (Patil et al., 2025) provide complete evalua-	885
837	<i>cessing Systems 2024, NeurIPS 2024, Vancouver, BC,</i>	tion code, and we follow their official settings. All	886
838	<i>Canada, December 10 - 15, 2024</i> .	baseline settings, including temperature, max steps,	887
839	The Use of Large Language	system prompts, sampling strategies, and tool for-	888
840	Models (LLMs) in Writing	mats, match the official evaluation code exactly.	889
841	An LLM (specifically OpenAI’s GPT-5 (OpenAI,	For factual reasoning datasets, the search tool is	890
842	2023)) was used solely for minor language editing,	implemented using the Google Search API, with	891
843	including grammar correction and light rephrasing	country set to “us” and top- <i>k</i> set to 5. We use only	892
844	for clarity. It did not contribute to the research	the text snippets returned by the API as observa-	893
845	design, and all scientific content is entirely the au-	tions, omitting full web browser outputs or long	894
846	thors’ own.	context summaries.	895
847	A More Implementation Details	BFCL-V4 and TAU2 are continuously evolving	896
848	Synthetic Data All personas used during data	agentic benchmarks, which replace older test data	897
849	synthesis are exclusively sourced from Persona	with more challenging samples. This makes pre-	898
		viously reported results potentially outdated. For	899

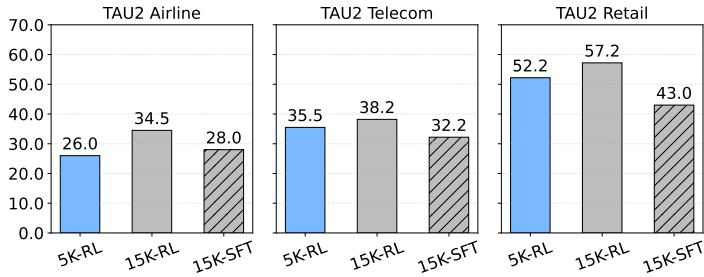


Figure 5: Impact of increased training data on RL performance, and comparison between RL and SFT at the same data scale.

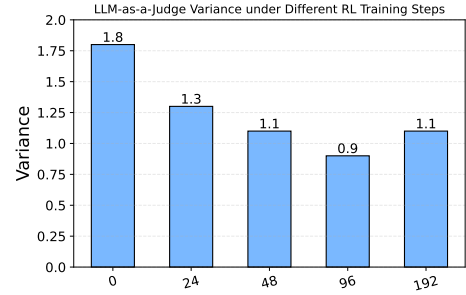


Figure 6: Stability analysis of LLM-as-a-Judge: score variance on the sample tasks across different training steps.

consistency, we re-evaluate all baselines using the latest BFCL-V4 and TAU2 versions available as of December 2025.

B Additional Experiment Results

Figure 5 illustrates how increasing training data affects agentic performance on the TAU-2 benchmark. Models trained with 15K tool-use samples clearly outperform those trained with 5K, highlighting the quality of our synthetic data and suggesting that scaling synthetic data and RL compute can further enhance agentic capabilities. The figure also compares RL and SFT under the same data scale. The SFT trajectories are synthesized by Qwen3-235B; however, SFT yields much smaller gains than RL, as RL can generate far more diverse trajectories through exploration, which is a crucial factor for improving agentic behavior.

During RL training, we use an LLM to assign rewards to trajectories based on our synthesized rubrics. These rubrics guide the evaluation process, substantially reducing LLM-as-a-Judge variance. We sample 100 tasks and generate trajectories for each task at different RL training steps, scoring them with Qwen30B-A3B-Instruct and computing the mean. Figure 6 reports the percentage score variance (repeat 6 times) across training steps. From the untrained model to near convergence, the LLM’s scores on the same tasks become increasingly consistent, demonstrating the stability of our rubric-based RL training.

C Synthetic Task and Tool Examples

In Table 6, we provide examples of the synthetic data, including task descriptions, user-only information, and the tool set formatted according to the OpenAI function-calling schema.

Table 7 presents the rubrics used to evaluate task completion, covering constraints, sub-goals, and interactions between the user and the agent.

Table 8 illustrates the complete execution flow of our model operating in simulated environments, where it invokes tools, interacts with the user, and solves problems throughout the process.

To ensure full reproducibility, we release the entire data-generation code, including all prompts used for synthesis, the synthetic dataset, and the reinforcement learning code based on this data. All resources are available at the following link: <https://anonymous.4open.science/r/SYNTHAGENT-68A4/>.

All prompts used in the paper can be found in the anonymous repository under the path `tool_use_data_synthesis/functions`.

D Contribution Summary

Our method has clearly defined boundaries from prior work on data synthesis and environment simulation, mainly in the following aspects: (1) agentic training data synthesis, (2) stable environment simulation for agentic RL training, and (3) rubric-based rewards. These components have rarely been studied in previous studies.

Prior data synthesis work (Wang et al., 2023) has focused primarily on reasoning tasks, largely overlooking diverse tool-use scenarios. We synthesize task-specific tool ecosystems with deliberately underspecified instructions containing information gaps and user-private context. This necessitates multi-turn communication and long-horizon planning, promoting procedural generalization to new tools rather than memorizing interfaces or following scripted steps.

Existing work on environment simulation, such as ToolLLM (Qin et al., 2024), primarily targets

Category	Benchmark / Subset	Test Size	Description
Agentic Tool Use	TAU-2: Airline (Barres et al., 2025) (Avg@4)	50	Airline booking and service workflows; multi-turn tool use with user interventions.
	TAU-2: Telecom (Avg@4)	114	Telecom troubleshooting and account operations; user can also call tools.
	TAU-2: Retail (Avg@4)	114	Retail returns and order management; multi-step tool execution in dialogs.
	BFCL-V4: Multi-turn / Base (Patil et al., 2025)	200	Multi-turn function calling across domains; end-to-end tool orchestration.
	BFCL-V4: Multi-turn / Miss Func	200	Missing or invalid functions; tests tool rejection and plan adjustment.
	BFCL-V4: Multi-turn / Miss Param	200	Missing required arguments; tests parameter elicitation and correction.
	BFCL-V4: Multi-turn / Long Context	200	Long-context dialogs; tests memory and consistency over many turns.
	Math Reasoning	AIME24 ¹ (Avg@6)	30
AIME25 ² (Avg@6)		30	2025 AIME I&II across major topics; evaluates out-of-domain math reasoning.
OlympiadBench (He et al., 2024)		674	Olympiad-level math problems; tests hard multi-step reasoning.
HMMT25 ³ (Avg@6)		30	Recent contest math problems; evaluates robustness on new distributions.
Search Reasoning	FRAMES (Krishna et al., 2025)	824	Search-based QA with evidence synthesis; evaluates factual reasoning under retrieval.
	xBench (Chen et al., 2025b)	100	Deep-search benchmark; multi-hop exploration and cross-source synthesis.
	WebWalker (Wu et al., 2025)	200	Web navigation and retrieval; multi-step searching in dynamic settings.

¹ https://huggingface.co/datasets/HuggingFaceH4/aime_2024

² <https://huggingface.co/datasets/math-ai/aime25>

³ https://huggingface.co/datasets/MathArena/hmmt_feb_2025

Table 5: Overview of evaluation benchmarks.

972 benchmark construction. In RL, however, reproducibility is critical: if the simulator returns different tool responses for the same state and action, 973 RL training becomes unstable. This issue is rarely 974 addressed. We introduce task-level finite mappings 975 that enforce consistent responses for identical tool 976 calls within each task, yielding stable simulations. 977

978 We automatically derive execution-aligned, task- 979 specific rubrics that cover subgoals, required inter- 980 actions, and disallowed behaviors, and use them 981 to build rubric-based RL rewards, an area rarely 982 explored in agentic RL training. 983

984 These elements are integrated in SYNTHAGENT 985 as a closed loop rather than a loose collection. Each 986 sample jointly specifies tools, instructions, hidden 987 information, and evaluation criteria, yielding a re- 988 producible agentic RL training recipe.

Fuzzy Task: Create a weekly digital newsletter for a senior resident that shares personal updates, meaningful stories, and family prompts in an accessible and heartfelt format.

User Only: The senior prefers speaking over typing and can comfortably record short voice messages of up to 90 seconds. The newsletter should reflect their voice and experiences, include a photo from their life, and feature a thoughtful quote related to family or memory. Each edition must end with a gentle question to invite responses from younger family members. The final version must be delivered as a well-formatted email that displays correctly on phones and tablets, with optional audio support for low-vision recipients.

Constraint: The agent must not use `VoiceTranscriber` on audio longer than 90 seconds. A violation occurs if the agent transcribes without confirming duration, assumes the audio is short enough, or proceeds when the audio is known/implied to exceed 90 seconds.

Tools (OpenAI function-calling spec):

VoiceTranscriber

Description: Converts short spoken audio clips into punctuated text; designed for seniors with a simple UI.

Parameters:

`audio_file_path` (string): File path or URL to the audio clip (.mp3/.wav).

LiteratureFinder

Description: Retrieves a short quote or poem excerpt from a curated public-domain database given a theme keyword.

Parameters:

`search_query` (string): Keyword/theme (e.g., family, memory, grandchild).

NewsletterFormatter

Description: Assembles text, image, and quote into an accessible HTML email template with responsive layout and an optional audio link.

Parameters:

`headline` (string): Main title/subject line of the newsletter.

`body_text` (string): Main content paragraph (e.g., weekly update).

`image_url` (string): URL to the featured image.

`quote` (string): Curated quote or poem excerpt to include.

`attribution` (string): Attribution formatted as "Author, Title".

`question_of_week` (string): Prompt question to encourage family responses.

`audio_narration_url` (string, optional): URL to an audio version of the newsletter.

ImageDescriber

Description: Generates accessibility-oriented alt-text for a given image URL.

Parameters:

`image_url` (string): URL of the image to be described.

AudioNarrationGenerator

Description: Simulates converting written text into an audio narration file for accessibility.

Parameters:

`text` (string): The text content to be narrated.

Table 6: An example of our synthesized tool-use training data, including the underspecified instruction (`fuzzy task`) and the OpenAI function-calling style tool specifications (`tools`).

Rubric

1. Forbidden Behaviors

Duration constraint: The agent must not use `VoiceTranscriber` on audio longer than 90 seconds. A violation occurs if the agent transcribes without confirming duration, assumes the audio is short enough, or proceeds when the audio is known/implied to exceed 90 seconds.

2. Task Sub-goals

- (1) Obtain a ≤ 90 s voice update and confirm duration.
 - (2) Transcribe the voice message via `VoiceTranscriber`.
 - (3) Retrieve a relevant quote via `LiteratureFinder`.
 - (4) Include a user-provided image and generate alt-text via `ImageDescriber`.
 - (5) Add a family prompt question.
 - (6) Generate audio narration via `AudioNarrationGenerator`.
 - (7) Format the newsletter via `NewsletterFormatter`.
-

3. Required User Interactions (must ask/confirm before acting)

- (1) Voice message availability, file path, and whether it is under 90 seconds.
 - (2) Image availability and a valid image URL.
 - (3) Newsletter theme/focus (memory/story to highlight).
 - (4) Audience and tone preferences.
 - (5) Whether audio narration is desired/needed.
-

Table 7: An example rubric for the synthesized training data instance in Table 6, including forbidden behaviors, task sub-goals, and required user interactions.

Role	Content
User	Create a weekly digital newsletter for a senior resident that shares personal updates, meaningful stories, and family prompts in an accessible and heartfelt format. (Ask for clarification if required parameters are missing.)
Assistant	Requests missing details (personal update, theme, image URL, audio narration, target audience).
User	The senior recorded a short voice message about planting flowers in the garden with a grandchild—would you like me to transcribe that as the personal update?
Assistant	Confirms using the voice message and asks for two required details: (i) audio length ≤ 90 seconds, and (ii) an .mp3 or .wav file link/path.
User	Yes, it is 85 seconds and here is the .mp3 link: https://example.com/audio/flower_garden.mp3 .
Assistant (Tool Call)	VoiceTranscriber(audio_file_path="https://example.com/audio/flower_garden.mp3")
Tool (Response)	{"transcribed_text": "Every spring, I would plant marigolds along the garden path... I feel her joy all over again.", "confidence_score": 0.96}
Assistant	Plans to add a complementary literary quote and calls LiteratureFinder with a relevant theme.
Assistant (Tool Call)	LiteratureFinder(search_query="garden and memory")
Tool (Response)	{"quote": "They flash upon that inward eye ...", "attribution": "-- William Wordsworth, from 'I Wandered Lonely as a Cloud'}
Assistant	Asks the user for a featured image URL to include in the newsletter.
User	Image link: https://example.com/images/marigolds-with-grandchild.jpg .
Assistant (Tool Call)	ImageDescriber(image_url="https://example.com/images/marigolds-with-grandchild.jpg")
Tool (Response)	{"description": "A sunny garden scene ... 'Grandma's Sunshine Patch.'", "success": true}
Assistant	Proposes a family prompt question and formats the newsletter using NewsletterFormatter.
Assistant (Tool Call)	NewsletterFormatter(headline="Planting Marigolds, Growing Memories", body_text="Every spring, ...", image_url="https://example.com/images/marigolds-with-grandchild.jpg", quote="They flash upon that inward eye ...", attribution="-- William Wordsworth, from 'I Wandered Lonely as a Cloud'", question_of_week="What's a garden, flower, or outdoor tradition that brings back warm childhood memories for you?")
Tool (Response)	Returns html_output and preview_url (omitted for brevity).
Assistant	Generates an optional audio narration for accessibility.
Assistant (Tool Call)	AudioNarrationGenerator(text="Planting Marigolds, Growing Memories. Every spring, ...")
Tool (Response)	{"narration_status": "success", "audio_url": "https://example.com/audio/narration/garden_memory_narrated.mp3"}
Assistant (Final)	<answer>The newsletter is generated (HTML preview available) and an audio narration link is provided for accessibility.</answer>

Table 8: An example of multi-turn interaction in our mock environment (user + tools) for the synthesized tool-use instance in Table 6, illustrating user interaction, tool calls, and tool responses.