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

Yuanjie Lyu[1], **Chengyu Wang**[2*], **Lei Shen**[3], **Jun Huang**[2], **Tong Xu**[1*]
[1] University of Science and Technology of China
[2] Independent Researcher
[3] Xi'an Jiaotong University
s1583050085@gmail.com

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 showing performance comparable to some larger baselines in certain domains.[1]

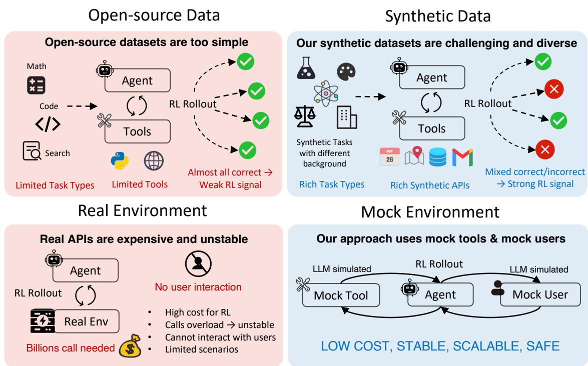

Figure 1: Comparison between existing agentic RL training recipes and ours. Open-source agentic training data are narrow in domain, while real-world APIs are costly and unstable. We replace these with diverse synthetic tasks and associated mock environments.

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 smaller models to reproduce the agentic capabilities of large models has become an important research direction (Yue et al., 2024; Cai et al., 2025; Lyu et al., 2025a; 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

*Corresponding authors.

[1]Code and prompts for data synthesis pipeline and RL training: https://github.com/haruhi-sudo/SYNTHAGENT.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12529–12545

with weak learning signals (Yu et al., 2025).

- *Absence of stable, diverse environments.* Real environments rarely support real-time model-user interaction and offer only a narrow tool set. RL rollout also requires a massive number of tool calls, making it impractical to rely on costly real-world APIs (LongCat, 2025).

To address these, we introduce SYNTHAGENT, a framework that synthesizes tool-use tasks along with lightweight mock tool interfaces. A strong agentic teacher LLM generates novel tasks and their associated tools, guided by diverse persona backgrounds (Ge et al., 2025). As shown in Figure 1, each task is paired with its own tool ecosystem, greatly expanding task and tool diversity. Moreover, the synthetic tools require no real deployment: an open-source LLM simulates both user and tool responses locally, ensuring stability.

Specifically, *for synthesizing training data*, we introduce an information gap by rewriting detailed workflows as underspecified instructions, while critical details are hidden in a private user context. This design forces agents to actively query users and call tools to recover missing information, encouraging genuine long-horizon interaction. Second, *for LLM-based tool response consistency*, we maintain a task-level mapping of prior tool calls and responses. New calls are answered by consulting this mapping for consistent replies. As each synthetic task has a unique toolset, the mapping is scoped per task, keeping it lightweight during rollout. Finally, *for reward design*, we avoid subjective LLM-written rubrics and derive rewards from observable behavior. Using the workflow from data synthesis as a reference, we extract corresponding high-level subgoals from real execution trajectories, each reachable via multiple valid paths. This yields execution-grounded rewards that support diverse strategies, while filtering out low-quality data when the teacher fails to reliably complete the workflow.

We evaluate our approach on 14 recent, challenging datasets spanning agentic tool use (Yehudai et al., 2025) and short-horizon reasoning. In real-world tasks, models trained on synthetic data within virtual environments substantially outperform those trained on open-source datasets. After training, our 8B–14B models outperform certain larger open-source baselines on multiple agentic benchmarks. In summary, the major contributions of this work are as follows:

- We introduce an open-source framework for synthesizing diverse agentic tool-use tasks, with stable, lightweight mock tool interfaces.
- In synthetic tasks that require genuine long-horizon interactions, we train models with execution-grounded, rubric-based rewards.
- Extensive experiments on 14 challenging datasets demonstrate that models trained on synthetic data and virtual environments achieve strong real-world performance.

2 Related Work

2.1 Agentic Reinforcement Learning

Recent studies show that RL outperforms SFT in long-horizon planning and adaptive decision-making (Zhang et al., 2025), making RL a core paradigm for training LLM agents in dynamic, multi-turn environments (Mialon et al., 2023). Classical methods such as Q-learning (Mnih et al., 2015), PPO (Schulman et al., 2017), and self-play (Silver et al., 2017) have provided the conceptual foundation for agentic optimization in LLM-based systems. These techniques have evolved into language-centric RL frameworks, where natural-language reasoning steps, tool calls, and observations are treated as latent states and actions (Yao et al., 2023; Zhang et al., 2025). Recent work has further improved RL algorithms to better couple exploration with robust tool use in long-horizon tasks, including verifiable-reward RL (Su et al., 2025), entropy-regularized policy optimization (Dong et al., 2025a), and agent-specific PPO/-GRPO variants (Dong et al., 2025c). Despite this progress, research remains largely focused on RL algorithms, with considerably less attention given to data and environment design.

2.2 Synthetic Data for Agentic Training

The effectiveness of agentic RL depends on high-quality, diverse data and environments, which remain scarce (Yehudai et al., 2025). Early works such as Self-Instruct (Wang et al., 2023) use strong but closed-source LLMs to generate instruction-following data for training smaller open-source models. To further increase diversity, Ge et al. (2025) propose Persona Hub, which curates one billion web-derived personas to enable diverse synthetic data generation. In parallel, Qin et al. (2025) identify a "rectified scaling law" for synthetic data:

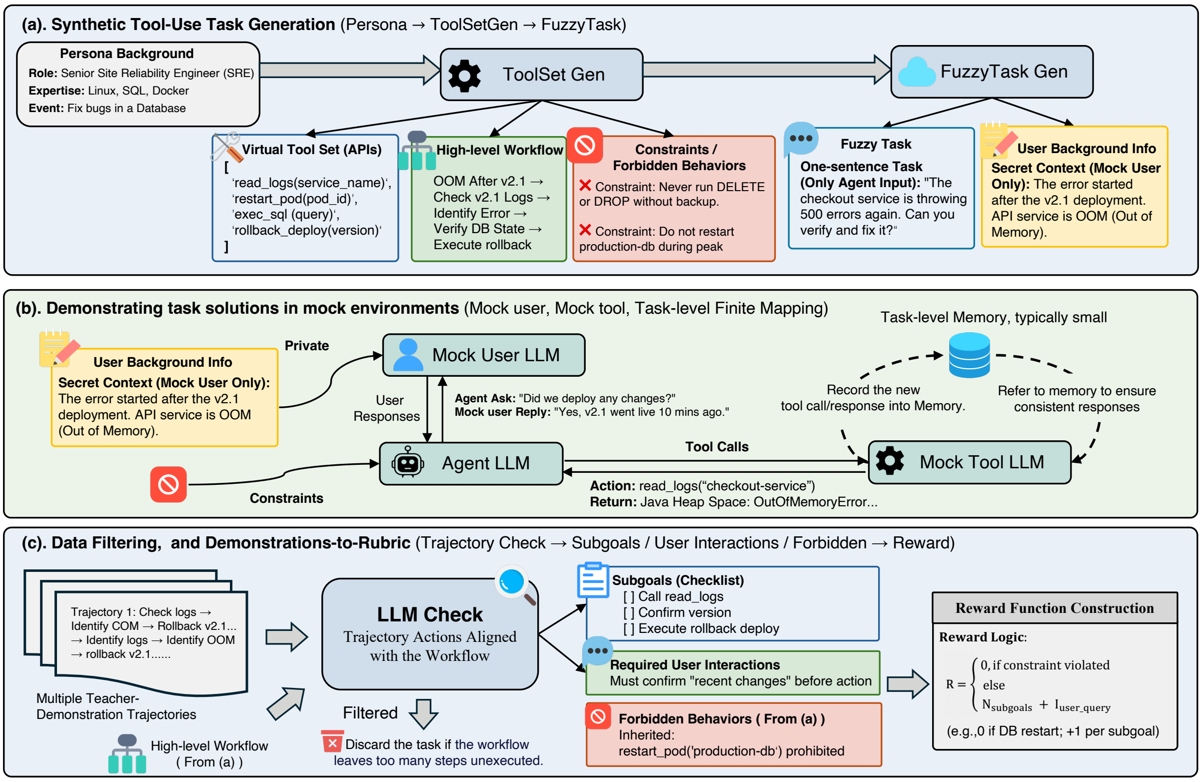

Figure 2: A unified pipeline for generating synthetic tool-use tasks, constructing stable mock environments, and deriving rubric-based rewards for agentic RL. Diverse tasks and tool ecosystems are created, guided by personas. For each synthetic task, an LLM-simulated user and environment are employed.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

as long as diversity is maintained, gains from synthetic pre-training persist even at very large scales.

For environment construction, benchmarks (Qin et al., 2024) rely on real-world APIs for authenticity or use LLMs to simulate existing APIs and reduce cost. However, these environments remain constrained by real services and suffer from limited diversity (Lu et al., 2025). Moreover, 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 nontrivial restrictions.[2]

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 $a^\star$. Consequently, rollouts $\tau \sim \pi_\theta$ are highly homogeneous. At many steps t, the policy $\pi_\theta(a \mid s_t)$ is nearly deterministic; thus, during RL training, the variation of the advantage under π_θ may become negligible:

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

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:

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

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:

$$s_0 \mapsto (I, C), \qquad \text{s.t. } H(a^\star \mid I) \gg \epsilon. \tag{3}$$

Initially, I is insufficient to determine the optimal action $a^\star$ uniquely; the conditional entropy $H(a^\star \mid I)$ 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 $H(a^\star \mid I)$, the policy must first query for the missing context C before invoking tools. As observations $o_{\le t}$ gradually reveal the hidden information, uncertainty decreases ($H(a^\star \mid I, o_{\le t}) < H(a^\star \mid I)$). Early decisions therefore become nontrivial: the agent must decide which clarification to ask first to elicit informative observations o_t, preventing $\pi_\theta(\cdot \mid s_t)$ 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 $\texttt{tool} + \texttt{args}$) 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 $\hat{g}_t = \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \hat{A}_t$ 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

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

[3]These 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.

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:

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

When the model issues a valid tool call u, the simulator examines similar entries in $\mathcal{M}$ and determines whether any are semantically equivalent. 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, and it 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 collected trajectories leave any workflow steps uncovered, 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,

$$R(\tau) = \mathbb{I}(\tau) \cdot \frac{\left(N_{\text{subgoals}}(\tau) + I_{\text{user\_query}}(\tau)\right)}{2}. \tag{5}$$

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. Table 2 further presents the complexity distribution and interaction length statistics of 3,000 newly synthesized tool-use instances, which are held out from training and used as a validation set.

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 Tool-Star (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 Qwen3-235B[4] with persona information (Ge et al., 2025) to rewrite each problem into a scenario-based variant. Each synthesized problem is then solved

[4]For 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.

Rubric score bin	Task percentage	Avg tool calls	Avg user interactions
$[0.00, 0.25)$	53.5%	4.2	2.9
$[0.25, 0.50)$	16.0%	4.4	2.8
$[0.50, 0.75)$	19.5%	3.7	2.7
$[0.75, 1.00]$	11.0%	3.5	2.4

Table 2: Complexity distribution and interaction length of 3,000 newly synthesized tool-use instances, held out from training. Complexity is measured by scoring Qwen3-8B's trajectories using the rubric.

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 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 are recent, with most introduced in or after 2025, 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

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 3: 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 4: 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**.

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. Among sampled raw tool-use instances, our filtering strategy in Section 3.3, based on whether teacher execution leaves workflow steps unexecuted, removes about 9.1% of examples. An audit of discarded cases identifies three main failure modes: missing critical parameters, where required inputs cannot be derived from the instruction or permitted user interactions; toolset–workflow mismatch, where available tools cannot support key workflow steps; and constraint–goal conflict, where synthesized constraints block actions needed to complete the task.

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 3 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 substantially improves the agentic performance of small models, making them competitive with stronger open-source baselines. Using synthetic tool-use tasks and fully simulated environments, our method yields substantial gains across real-world agentic benchmarks. On TAU-2 and BFCL-Multi-turn, SYNTHAGENT-8B scores 42.9 on average (+12.3 over Qwen3-14B, and +6.9

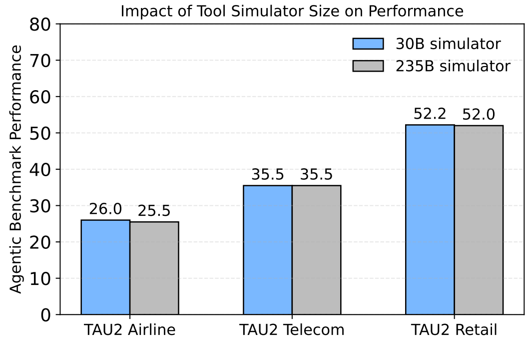

Figure 3: Effect of tool-simulator size on TAU-2 performance (5,000 training samples), showing negligible gains from larger simulators.

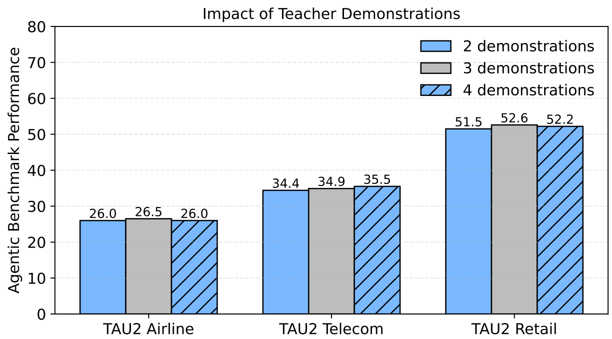

Figure 4: Influence of teacher-demonstration count when constructing task-level rubrics (5,000 training samples), indicating that additional demonstrations yield limited gain.

over Qwen3-32B). The improvements are even greater with SYNTHAGENT-14B, which scores 46.3 on average. The results demonstrate that our synthetic training strategy can close the gap with much larger models in certain domains. Still, it does not indicate that small models can match or replace larger models in general capability.

Synthetic data and simulated environments substantially outperform existing open-source datasets. Open-source agentic datasets cover only a limited set of tools and cannot capture the complexity of real tool-use logic. For example, in the latest ToolStar (Dong et al., 2025b) dataset, RL-trained models show only limited improvement in agentic performance. In contrast, SYNTHAGENT-8B achieves great performance in TAU-2, consistently surpassing open-source baselines, with BFCL-Multi-turn showing the same pattern. These results indicate that diverse and challenging synthetic tasks are far more effective for strengthening a model's agentic abilities.

Beyond tool-use tasks, the model also generalizes to short-horizon reasoning. Table 4 shows that, although math and search tasks are not the primary focus of our training (only 4,000 instances sampled from ToolStar are included), SYNTHAGENT still achieves substantial gains. Under the same non-thinking setting, SYNTHAGENT-8B significantly outperforms Qwen3-14B and even exceeds an 8B model trained on 30,000 ToolStar examples. The improvements for SYNTHAGENT-14B are even larger. These results indicate that our method transfers effectively to new reasoning domains, with tool-use data also exhibiting strong generalization in reasoning tasks.

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

Method	TAU-2 Bench			BFCL-V4 Multi-turn
	Airline	Telecom	Retail	Avg.
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 5: Ablation study on agentic benchmarks.

4.3 Further Analysis

Ablation The results in Table 5 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 achieving competitive performance relative to larger baselines.

Limitations

Agentic training data synthesis is an increasingly important research topic. Technical reports from leading foundation models (Qwen3 (Yang et al., 2025), LongCat (LongCat, 2025), Kimi K2 (Bai et al., 2025), DeepSeek V3.2 (DeepSeek-AI, 2025), Minimax M2 (Chen et al., 2025a)) consistently show that synthetic data, rather than real-world data, forms the core of agentic RL. However, these models do not release their synthetic datasets or detailed synthesis procedures, which restricts our ability to refine the SYNTHAGENT pipeline and compare against stronger baselines. Moreover, SYNTHAGENT depends on a strong teacher model during synthesis. Although filtering based on real teacher executions improves data quality, synthesized tasks may still not fully capture the complexity of real-world agentic settings. Future work should explore more robust synthesis strategies, reduce dependence on strong teachers, and extend the pipeline to richer settings such as multi-agent interaction (Lyu et al., 2025b).

Ethical Considerations

As an open-source framework for synthetic data generation and agentic RL training, our work poses relatively limited direct risk. However, biases from teacher models or persona sources may propagate into synthesized tasks. We recommend careful monitoring and bias evaluation before using such systems in sensitive tasks.

Acknowledgments

This work was supported in part by the grants from National Science and Technology Major Project (No. 2023ZD0121104), and the Anhui Natural Science Foundation (No. 2508085ZD006).

References

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

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*.

Wenrui Cai, Chengyu Wang, Junbing Yan, Jun Huang, and Xiangzhong Fang. 2025. Enhancing reasoning abilities of small llms with cognitive alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7434–7449.

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

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou, Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng

Liu, Jinpeng Li, Kangyi Liu, and 14 others. 2025b. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *arXiv preprint arXiv:2506.13651*.

DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025a. Agentic entropy-balanced policy optimization. *arXiv preprint arXiv:2510.14545*.

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

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

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2025. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

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

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

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, and 11 others. 2025. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2508.13167*.

Team Meituan LongCat. 2025. Longcat-flash technical report. *arXiv preprint arXiv:2509.01322*.

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 *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1160–1183.

Yuanjie Lyu, Chengyu Wang, Jun Huang, and Tong Xu. 2025a. From correction to mastery: Reinforced distillation of large language model agents. *arXiv preprint arXiv:2509.14257*.

Yuanjie Lyu, Chao Zhang, Yuhao Chen, Yong Chen, and Tong Xu. 2025b. Streamlining the collaborative chain of models into a single forward pass in generation-based tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6390–6404.

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. *arXiv preprint arXiv:2505.07773*.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

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, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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 *Forty-second International Conference on Machine Learning*.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

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. Scaling laws of synthetic data for language models. *arXiv preprint arXiv:2503.19551*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

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. *arXiv preprint arXiv:2402.03300*.

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. *arXiv preprint arXiv: 2409.19256*.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan 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. *arXiv preprint arXiv:1712.01815*.

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. *arXiv preprint arXiv:2503.23829*.

Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral cloning from observation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4950–4957. International Joint Conferences on Artificial Intelligence Organization.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. Webwalker: Benchmarking llms in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*.

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. *Science China Information Sciences*, 68(2):121101.

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, and 40 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. 2025. Survey on evaluation of llm-based agents. *arXiv preprint arXiv:2503.16416*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yuanhao Yue, Chengyu Wang, Jun Huang, and Peng Wang. 2024. Distilling instruction-following abilities of large language models with task-aware curriculum planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6030–6054.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Yue Liao, Hongru Wang, Mengyue Yang, and 6 others. 2025. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark W.

Barrett, and Ying Sheng. 2024. Sglang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

The Use of Large Language Models (LLMs) in Writing

An LLM (specifically OpenAI's GPT-5 (OpenAI, 2023)) was used solely for minor language editing, including grammar correction and light rephrasing for clarity. It did not contribute to the research design, and all scientific content is entirely the authors' own.

A More Implementation Details

Synthetic Data All personas used during data synthesis are exclusively sourced from Persona Hub (Ge et al., 2025). During toolset generation, we define virtual tools following the OpenAI function-calling specification and filter out any samples that cannot be parsed into a valid function-calling format. In total, we generated 15,096 tool-use tasks, along with a smaller set of math or search reasoning tasks (approximately 4,000). All data were produced locally with open-source LLMs, without reliance on commercial APIs, ensuring a stable and cost-efficient pipeline.

We analyze 3,000 newly synthesized tool-use instances held out from training, and report rubric-based scores from Qwen3-8B trajectories for each persona category. Persona categories are drawn from a predefined domain set, and Qwen3-235B is used to assign each persona to one category. Table 6 shows that the low-score rate (< 0.25), which we use as a proxy for difficult but RL–useful data, is broadly similar across categories, suggesting that task difficulty is not dominated by a particular persona type.

Benchmarks Detailed benchmark statistics (e.g., subsets and test sizes) are reported in Table 7. Agentic tool-use benchmarks (TAU-2 and BFCL-V4 Multi-turn) measure long-horizon, multi-turn interaction with unfamiliar tools. Reasoning benchmarks (math and search) evaluate out-of-domain generalization, where tasks typically involve fewer tool calls but require deeper multi-step reasoning and evidence synthesis.

Training We perform reinforcement learning using the GRPO algorithm (Shao et al., 2024) within the VERL framework (Sheng et al., 2024). We use a global batch size of 128, a PPO mini-batch size of 16, a rollout size of 16, and a maximum response length of 13,000 tokens, training for 2 epochs on 8×NVIDIA H20 GPUs. The clipping ratio is constrained between 0.2 and 0.28, and we allow up to 16 turns per rollout. Since Qwen3 base models already exhibit strong tool/function-calling capabilities and reliably follow the required format, we skip the SFT phase for format learning and directly proceed with RL training.

Inference During inference, the model was deployed with SGLang (Zheng et al., 2024) to increase throughput. TAU-2 (Barres et al., 2025) and BFCL (Patil et al., 2025) provide complete evaluation code, and we follow their official settings. All baseline settings, including temperature, max steps, system prompts, sampling strategies, and tool formats, match the official evaluation code exactly. For factual reasoning datasets, the search tool is implemented using the Google Search API, with country set to "us" and top-k set to 5. We use only the text snippets returned by the API as observations, omitting full web browser outputs or long context summaries.

BFCL-V4 and TAU2 are continuously evolving agentic benchmarks, which replace older test data with more challenging samples. This makes previously reported results potentially outdated. For 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 vari-

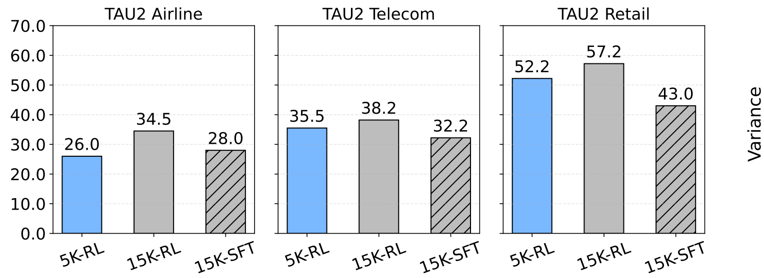

Figure 5: Impact of increased training data on RL performance, and comparison between RL and SFT at the same data scale.

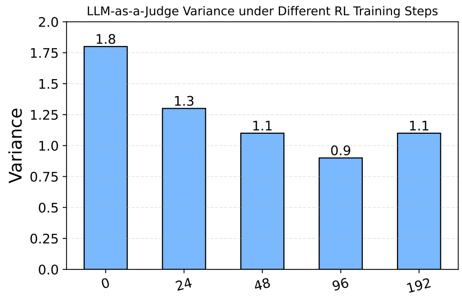

Figure 6: Stability analysis of LLM-as-a-Judge: score variance on the sample tasks across different training steps.

Metric	Legal&Gov.	Health	Technology	Education	Media/Arts/Sports	Business&Work	Others
#Tasks	810	297	135	708	773	293	111
% of all	25.9%	9.5%	4.3%	22.6%	24.7%	9.4%	3.6%
Low-score rate (< 0.25)	55.0%	57.6%	49.1%	51.9%	53.9%	57.6%	53.7%

Table 6: Domain distribution and low-score rate across persona categories on 3,000 newly synthesized tool-use instances held out from training. Scores are rubric-based and computed from Qwen3-8B's trajectories, judged by Qwen3-235B. A low score (< 0.25) indicates relatively hard, but potentially RL-beneficial, training instances.

ance. We sample 100 tasks and generate trajectories for each task at different RL training steps, scoring them with Qwen3-30B-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 8, 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 9 presents the rubrics used to evaluate task completion, covering constraints, sub-goals, and interactions between the user and the agent.

Table 10 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://github.com/haruhi-sudo/SYNTHAGENT`.

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

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	2024 AIME math problems in algebra and geometry; assesses advanced reasoning.
	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 7: Overview of evaluation benchmarks.

We automatically derive execution-aligned, task-specific rubrics that cover subgoals, required interactions, and disallowed behaviors, and use them to build rubric-based RL rewards, an area rarely explored in agentic RL training.

These elements are integrated in SYNTHAGENT as a closed loop rather than a loose collection. Each sample jointly specifies tools, instructions, hidden information, and evaluation criteria, yielding a reproducible 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 8: 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 $\leq$90s 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 9: An example rubric for the synthesized training data instance in Table 8, 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 $\leq$ 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 10: An example of multi-turn interaction in our mock environment (user + tools) for the synthesized tool-use instance in Table 8, illustrating user interaction, tool calls, and tool responses.