COMPOWORLD: COMPOSITIONAL ENVIRONMENT SCALING FOR GENERAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Automatically generated environments provide a scalable source of interaction data for training general agents. However, existing approaches mainly generate tasks within a single environment, while real-world workflows require agents to connect information and actions across multiple services. We introduce Compositional Environment Scaling (**CompoWorld**), which expands the task space by composing a finite library of reusable services. Coding agents turn tool specifications into verified services with typed states and shared interfaces, while a world model handles tools that cannot be reliably implemented. A random-walk procedure connects services through dependency graphs, enabling the generation and verification of tasks that require information to flow across services. Verified trajectories support supervised fine-tuning (SFT), while our Completion-Focused Rubric Reward guides reinforcement learning (RL) toward full task completion by emphasizing criteria with lower pass rates within each rollout group. We construct 448 services exposing 10,130 tools and use 3K SFT trajectories and 1K RL tasks to train Qwen3.6-35B-A3B. Experimental results show that CompoWorld improves on its backbone by 9.17 points on average across eight benchmarks. On AutomationBench, it surpasses frontier models such as Claude Opus 4.6 and leads all compared agent-specialized 35B-A3B models.

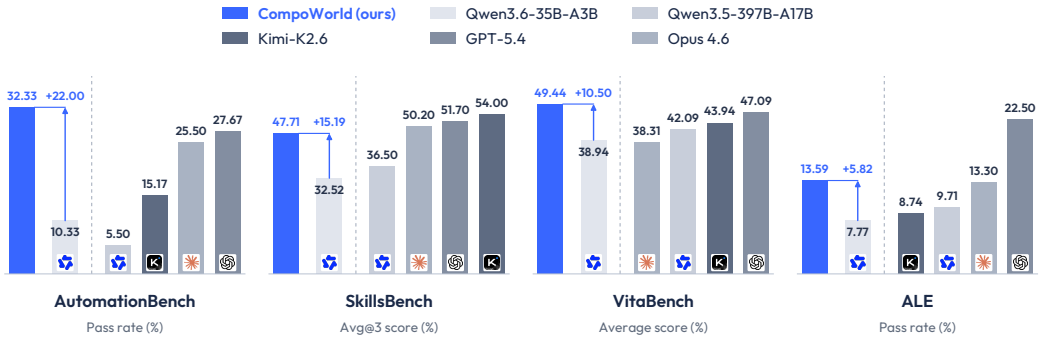


Figure 1: Performance of CompoWorld and five foundation models on four challenging agent benchmarks.

1 INTRODUCTION

Large language models (LLMs) are evolving from text generators into agents that reason, use tools, and act in digital worlds (Yao et al., 2023; Schick et al., 2023). Training these agents requires more than static demonstrations: it requires interactive environments in which actions change external states and task outcomes can be evaluated. A recent survey identifies environment synthesis and evaluation as central components of agent learning (Li et al., 2026a). This motivates **environment scaling**: expanding the diversity of environments and verifiable tasks available for training. For example, AgentScaler (Fang et al., 2025), ScaleEnv (Tu et al., 2026), and Agent-World (Dong et al., 2026) pursue this direction through the automated construction of tool-interaction environments and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tasks. Together, these efforts establish environment diversity as a promising axis for improving agent capabilities and enabling transfer to unseen tasks.

Beyond expanding the collection of environments, an equally important question is how to scale the dependencies between them. Real-world workflows often couple states across multiple systems: an agent may query a Snowflake database for overdue tickets, consult a PDF operating manual to determine the required response, and notify managers and customers by email. Success depends on carrying the right information and constraints across these systems, rather than merely making more tool calls. Multi-application benchmarks such as AppWorld (Trivedi et al., 2024) already expose this requirement, and Terminal-Universe (Wu et al., 2026a) explores cross-workspace tasks spanning related codebases. We target a complementary abstraction: independently executable services that can be reused and recombined through task-specific causal dependencies. This makes the composition of services itself a controllable dimension of environment scaling.

We introduce Compositional Environment Scaling (**CompoWorld**), a framework that constructs cross-environment tasks from reusable execution services. Each service owns a typed state, exposes a tool set, and defines its transition logic. Typical services include email, Slack, calendars, and database applications. CompoWorld composes these services according to the dependencies required by a task, allowing a finite service library to support a much larger space of workflows. The objective is to train agents to coordinate familiar services in novel combinations, thereby linking environment construction to the broader problem of compositional generalization (McCurdy et al., 2024). Compositional environment scaling introduces three challenges. First, automatically generated services must execute reliably on their own while remaining compatible with other services. Second, synthesized tasks must contain meaningful cross-service dependencies and remain solvable and verifiable. Third, these tasks must provide useful supervision for both supervised fine-tuning (SFT) and agentic reinforcement learning (RL), including when an agent completes only part of a workflow.

To address the first challenge, CompoWorld standardizes service states and interaction interfaces using typed Python schemas. We collect tool specifications from public Model Context Protocol (MCP) implementations and use coding agents within a harness to build executable mock services. For long-tail tools that cannot be implemented reliably, a world model serves as a simulator. This hybrid design preserves each service's full tool interface, allowing it to participate in composed workflows even when some operations are simulated. To address the second challenge, CompoWorld uses a random-walk procedure to sample services and connect them into a service-level dependency graph. Each node represents an independently executable service, while each directed edge indicates that information or state from the source service is required by the target service. The graph captures dependencies without prescribing a fixed tool-call sequence. Given the selected services and graph, a task-generation agent instantiates initial states, goals, and constraints, while verification probes test the reachability of required transitions and the satisfiability of success conditions. This generation-and-verification process is itself agentic. To address the third challenge, CompoWorld uses verified successful trajectories for SFT and task rubrics for RL. Cross-environment tasks require multiple conditions to hold jointly, so partial workflow completion may still fail to satisfy the user's goal. Rubric rewards provide graded credit for satisfied conditions even when no rollout fully succeeds. We further introduce Completion-Focused Rubric Reward, which assigns higher weights to criteria with lower pass rates within each group. Combined with Group Relative Policy Optimization (GRPO) (Shao et al., 2024), this reward focuses on unmet requirements and promotes full task completion.

We construct 448 reusable services exposing 10,130 tools, and train Qwen3.6-35B-A3B with 3K SFT trajectories and 1K RL tasks. Evaluation spans eight challenging agent benchmarks, with an average gain of 9.17 points over the backbone. On AutomationBench, CompoWorld reaches a task success rate of 32.33% (+22.00 points), exceeding GPT-5.4 (27.67%) and Claude Opus 4.6 (25.50%) and approaching DeepSeek-V4-Flash (36.33%). It also leads all six compared agent-specialized 35B-A3B models on this benchmark. These results highlight the value of our training approach for cross-service workflows. Figure 1 shows results on a subset of these benchmarks.

2 Preliminaries and Formulation

General Agent Tasks. We model each LLM-agent environment as a partially observable Markov decision process (POMDP) without a task-specific reward function: $E_i = (S_i, A_i, T_i, \Omega_i, O_i)$, where S_i, A_i, and Ω_i denote the state, action (tool), and observation spaces, respectively, while

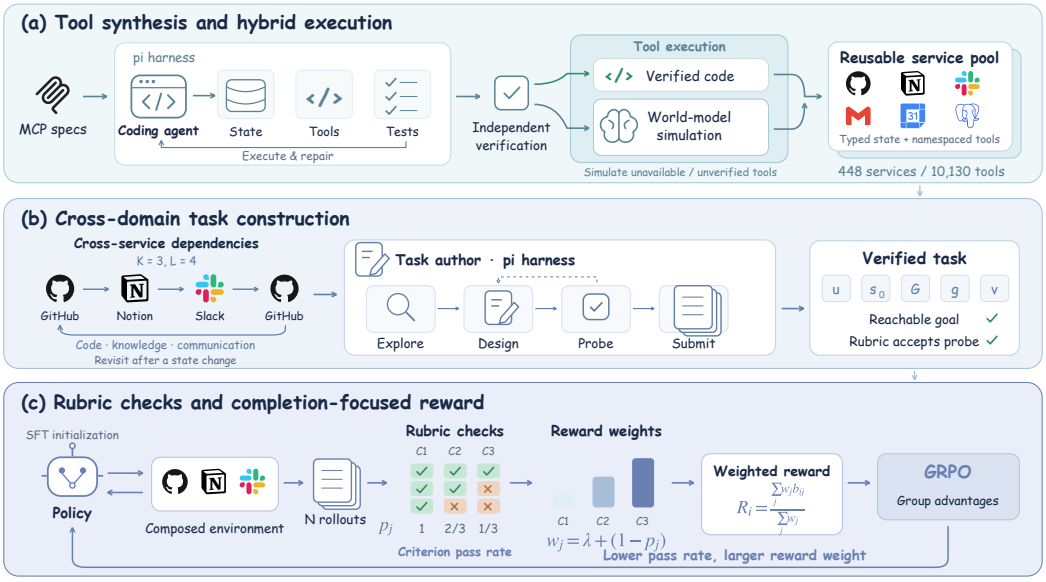


Figure 2: **CompoWorld overview.** (a) Verified tools with selective world-model simulation; (b) cross-service task generation and verification; (c) completion-focused rubric rewards for agent training.

$T_i(s' \mid s, a)$ and $O_i(o \mid s', a)$ define the transition and observation functions. Given an instruction u, the agent samples $a_t \sim \pi(\cdot \mid h_t)$ from the interaction history $h_t = (u, o_0, a_0, \ldots, a_{t-1}, o_t)$, after which $s_{t+1} \sim T_i(\cdot \mid s_t, a_t)$ and $o_{t+1} \sim O_i(\cdot \mid s_{t+1}, a_t)$. An interaction produces a trajectory $\rho = (o_0, a_0, o_1, \ldots, a_{H-1}, o_H)$, whose success is determined by a terminal verifier rather than per-step rewards.

Compositional Environments. Given independently executable environments $C = \{E_1, \ldots, E_K\}$, their composition is the product environment $E_C = \bigotimes_{E_i \in C} E_i$, with joint state space $S_C = \prod_{i=1}^{K} S_i$ and namespaced action space $A_C = \bigcup_{i=1}^{K} \{(i, a) \mid a \in A_i\}$. The namespace distinguishes otherwise identical tools. Observations are likewise associated with the invoked service, with $\Omega_C = \bigcup_{i=1}^{K} \{(i, o) \mid o \in \Omega_i\}$. For a joint state $s = (s_1, \ldots, s_K)$, action (i, a) updates only E_i and returns its local observation:

$$T_C(s' \mid s, (i, a)) = T_i(s'_i \mid s_i, a) \prod_{j \neq i} \mathbf{1}[s'_j = s_j], \quad O_C((k, o) \mid s', (i, a)) = \mathbf{1}[k = i] O_i(o \mid s'_i, a). \tag{1}$$

Thus, composition preserves each environment's local dynamics and observation function; information passes between environments through the agent's observations and subsequent actions.

3 CompoWorld: Compositional Environment Scaling

We introduce **CompoWorld**, a framework for scaling general agent training environments along a compositional dimension. As shown in Figure 2, the framework contains three parts. The first part covers how we build individual services as building blocks of compositional environments. The second part addresses the generation and verification of complex cross-environment tasks. The third part focuses on how we use these tasks for agent training.

3.1 Agent Environment Generation

Typed Environment Modeling. We first collect machine-readable MCP specifications through web crawling, retain those corresponding to relatively self-contained applications, and normalize them into a unified function-calling format. For each service, the specification provides tool names, descriptions, and typed parameter schemas, thereby defining the action space A_i, but leaves the state

space S_i, transition function T_i, and observation function O_i unspecified. A coding agent infers the service entities and constructs a Pydantic model for the environment state $s_i \in S_i$. Each entity is represented as a typed record, and the service state comprises the corresponding record collections. Pydantic validation rejects unknown fields and invalid updates, ensuring that state transitions conform to the service schema. Given this state model, the coding agent implements each tool $a \in A_i$ as an operation over the current state. Each invocation produces an updated state and a structured observation. Using the formulation in Section 2, the joint distribution of these outputs is

$$\Pr(s_i', o \mid s_i, a) = T_i(s_i' \mid s_i, a)\, O_i(o \mid s_i', a), \qquad s_i' \in S_i, \quad o \in \Omega_i. \tag{2}$$

For a deterministic implementation, the same state and tool invocation yield the same output pair, so both factors assign probability one to the returned values. This interface makes the state update and the returned observation explicit, allowing subsequent tools to operate on the updated records. All tools return responses in a unified JSON format, allowing independently generated services to share a common execution interface.

Agentic Synthesis and Verification. For each service, the coding agent, operating through the `pi` harness (Earendil Works, 2026) in an isolated execution sandbox, generates the state model, tool implementations, and a corresponding test suite. It iteratively executes the tests and repairs its implementation until both basic successful-use cases and error-handling cases pass. Because self-generated tests may inherit blind spots from the implementation, we subsequently conduct an independent validation pass. A separate agent session derives adversarial test cases directly from the tool specifications, including checks for boundary conditions. Tools that fail these checks are quarantined and are not incorporated as verified deterministic transitions.

Selective World-Model Simulation. Some tools cannot be faithfully implemented as deterministic local code, particularly those that depend on external systems or require functionality beyond the coding agent's capabilities. We retain support for such tools through an LLM-based world model, motivated by evidence that language world models can provide useful environment simulation for agent training (Zuo et al., 2026). Given the current state and a tool invocation, the model predicts the output pair (s_i', o) in Eq. 2, including both the required state update and the resulting observation. The predicted updates are instantiated through the same Pydantic models used by the deterministic tools, ensuring that they remain type-valid and consistent with the service state. The updated state is then available to later tool calls, including those handled by deterministic code. Each generated environment therefore combines verified deterministic implementations with selective world-model simulation for tools whose dynamics cannot be reliably encoded. Appendix F discusses environment quality, the limited scope of simulation, and illustrative cases.

The pipeline produces 448 reusable services, each with its own persistent state and namespaced interface, exposing 10,130 tools in total. These services share execution conventions and can be composed through the product construction $E_C = \bigotimes_{E_i \in C} E_i$. Appendix A describes the corpus and its domain distribution.

3.2 Cross-Environment Task Generation and Verification

The executable services provide the building blocks for *cross-environment tasks*. A task is represented as $x = (u, s_0, G, g, v)$, where u is the instruction, $s_0 \in S_C$ is the initial joint state, $G = (C, D)$ records service-level dependencies, $g \in S_C$ is a reachable reference goal state, and v verifies task-relevant outcome conditions.

Composition and Difficulty Control. To build a task, we sample a set C of K services from the pool and form the product environment E_C defined in Section 2, where tools are grouped by service namespace. Since E_C only defines which services are available together, we create the dependency graph $G = (C, D)$ using an environment walk. For a walk $(E_{i_1}, \ldots, E_{i_L})$, the length L counts service visits, including revisits. We define its edges and constraints as

$$\begin{aligned} D &= \{(E_{i_\ell}, E_{i_{\ell+1}}) \mid 1 \le \ell < L\}, \\ \{E_{i_1}, \ldots, E_{i_L}\} = C, &\qquad i_\ell \neq i_{\ell+1}\ (1 \le \ell < L), \qquad L > K. \end{aligned} \tag{3}$$

These constraints require the walk to cover every selected service, avoid consecutive visits to the same service, and revisit at least one service. Each edge represents an information dependency,

meaning that the input needed by service E_j can only be obtained from the observation returned by the preceding service E_i, while a revisit is required when an intervening step makes a value from the first visit stale and forces the agent to read it again. The walk guides task construction, while G records service-level dependencies rather than a unique tool-call sequence. The main difficulty control is L. Increasing L adds more linked service visits without requiring more distinct services. Other controls include the sampled domain and the number of services K. Together, these controls set the breadth and realism of the task, while the walk provides a basic structure that the authoring agent turns into a clear scenario.

Agentic Task Generation. A coding agent within the `pi` harness creates each task in the composed environment E_C through four phases within a bounded revision loop. In EXPLORE, the agent interacts with the services in an isolated sandbox to learn their tools and state formats. In DESIGN, it turns the sampled walk into a concrete scenario, constructs a validated initial state s_0, and drafts the user instruction u together with a structured rubric of objective criteria. In PROBE, the agent solves the task by executing the walk in E_C. The resulting joint state defines the goal g and serves as the reference answer. Each rubric criterion is translated into an executable final-state check, and the resulting verifier v must accept the reference solution. For a probe trajectory ρ with horizon H, this requires

$$s_0 \xrightarrow[E_C]{\rho} s_H = g, \qquad v(s_H) = 1. \tag{4}$$

The first condition establishes that the reference goal is reachable through actual tool calls, and the second checks that the rubric accepts this outcome. The verifier checks task-relevant conditions without requiring exact equality to g, allowing other successful trajectories to use different tool sequences and differ in unrelated state fields. Probing must also successfully exercise every tool used by the task. In SUBMIT, the agent finalizes u and the task artifacts.

3.3 General Agent Training

We first initialize the policy through supervised fine-tuning on verified successful trajectories, then train it through reinforcement learning in the composed environments. The task rubrics provide the reward signal, while the policy's performance on each criterion determines its contribution to the reward.

Completion-Focused Rubric Reward. Cross-environment success requires multiple conditions to hold together, while binary rewards give no credit for partial progress. For each task x, we sample N trajectories $\{\rho_i\}_{i=1}^{N}$ from $\pi_{\theta_{\text{old}}}$, each starting from an independent copy of s_0 in E_C. Let $b_{ij} \in \{0, 1\}$ indicate whether trajectory i satisfies criterion j among the task's M criteria, as checked against the final joint state and relevant recorded outputs. Uniform rubric averaging, $R_i^{\text{uniform}} = M^{-1} \sum_{j=1}^{M} b_{ij}$, provides partial credit, whereas full success requires $\text{Success}(\rho_i) = \prod_{j=1}^{M} b_{ij} = 1$. These outcome rewards require neither a prescribed tool sequence nor intermediate states matching a reference trajectory.

A high average rubric score can nevertheless leave the user's goal unmet, such as updating a record without sending the required notification. Uniform averaging rewards each criterion equally, regardless of how reliably it is satisfied. Our *Completion-Focused Rubric Reward* instead emphasizes criteria with lower pass rates in the current rollout group:

$$p_j = \frac{1}{N} \sum_{i=1}^{N} b_{ij}, \qquad w_j = \lambda + (1 - p_j), \qquad R_i = \frac{\sum_{j=1}^{M} w_j b_{ij}}{\sum_{j=1}^{M} w_j}, \tag{5}$$

where p_j is the group pass rate and $\lambda > 0$ retains a positive weight for every criterion. Weights are shared across the group, recomputed for each new group, and held fixed during the policy update. Normalization ensures $R_i \in [0, 1]$, with $R_i = 1$ only when all criteria pass. Satisfying a less frequently completed criterion earns more reward, focusing learning on remaining completion gaps while preserving partial credit. Reweighting cannot distinguish trajectories on a criterion that every rollout fails. Appendix C provides the gradient analysis.

We optimize the policy with GRPO (Shao et al., 2024) using the completion-focused reward in Eq. 5. Only agent-generated tokens contribute to the loss; tool observations provide context. Appendix B gives the full objective and normalization details.

Table 1: Main results on eight challenging agent benchmarks. Green values show score-point gains over Qwen3.6-35B-A3B.

Model	τ^3-Banking	Deep Planning	Vita Bench	Vita Bench2.0	Automation Bench	WildClaw Bench	Skills Bench	ALE
Frontier Closed-Source Models								
GPT-5.4	28.52	53.96	47.09	45.99	27.67	58.00	51.70	22.50
Claude Opus 4.6	20.27	55.21	38.31	37.71	25.50	54.60	50.20	13.30
Gemini-3.1 Pro	23.71	47.08	52.69	49.53	28.17	38.70	60.80	17.10
Open-Weight Models								
DeepSeek-V4-Flash	30.34	54.79	56.31	41.12	36.33	47.67	53.75	17.48
GLM-5.2	28.87	50.00	50.26	45.32	26.33	54.20	62.10	22.33
Kimi-K2.6	19.93	43.12	43.94	44.30	15.17	29.40	54.00	8.74
Qwen3.8-27B	33.68	52.92	41.84	47.23	38.50	56.20	35.57	20.40
Qwen3.5-397B-A17B	16.15	35.83	42.09	38.41	5.50	40.37	36.50	9.71
Qwen3.6-35B-A3B	10.65	26.04	38.94	34.47	10.33	44.28	32.52	7.77
Agent-Specialized Models (35B-A3B)								
Apodex 1.1 Mini	12.71	34.17	44.06	35.29	14.17	-	32.29	-
Occamy-1.0	37.10	-	41.75	-	27.60	49.16	-	-
Agents-A1	7.20	-	37.00	-	2.20	30.73	-	-
Nex-N2-mini	25.80	-	26.25	-	5.70	30.31	-	-
BigBang-1.0	10.30	-	46.00	-	14.80	32.87	-	-
Ornith-1.5-35B	21.70	-	40.25	-	18.50	45.91	-	-
CompoWorld (ours)	16.49	35.21	49.44	36.09	32.33	47.46	47.71	13.59
Δ vs. backbone	+5.84	+9.17	+10.50	+1.62	+22.00	+3.18	+15.19	+5.82

4 Experiments

We evaluate whether training on composed environments improves agent performance across domains, how these gains compare with stronger and similarly sized models, and how performance changes with the number of training environments.

4.1 Experimental Settings

Baselines. We compare CompoWorld with frontier foundation models and agent-specialized models. **Frontier closed-source models** include GPT-5.4 (OpenAI, 2026), Claude Opus 4.6 (Anthropic, 2026), and Gemini-3.1 Pro (Google DeepMind, 2026). **Open-weight foundation models** include DeepSeek-V4-Flash (0731) (DeepSeek-AI, 2026), GLM-5.2 (GLM-5 Team et al., 2026), Kimi-K2.6 (Moonshot AI, 2026), Qwen3.8-27B (Qwen Team, 2026c), Qwen3.5-397B-A17B (Qwen Team, 2026a), and Qwen3.6-35B-A3B (Qwen Team, 2026b). We additionally compare agent-specialized models at the 35B-A3B scale: Apodex 1.1 Mini (Apodex Team, 2026), Occamy-1.0 (Accio Team, 2026), Agents-A1 (Bai et al., 2026), Nex-N2-mini (Nex AGI, 2026), BigBang-1.0 (The BigBang Team, 2026), and Ornith-1.5-35B (Ornith, 2026). We evaluate Apodex under our protocol and use the source-reported results in Table 5 of the Occamy report (Accio Team, 2026) for the other five models. Dashes denote unavailable or incompatible results. For benchmarks that require an LLM judge, we use Qwen3.5-397B-A17B in some of our evaluations. Unless otherwise specified, our evaluations enable thinking with a high reasoning budget.

Challenging General Agent Benchmarks. We evaluate on eight challenging benchmarks spanning knowledge-grounded interaction, sustained planning, and workflow execution. τ^3-Banking (Shi et al., 2026) evaluates banking customer support requiring knowledge retrieval and policy-compliant tool use. WildClawBench (Ding et al., 2026) evaluates real-world workflows in native agent harnesses, and SkillsBench (Li et al., 2026b) assesses agents using reusable skills. AutomationBench (Shepard & Salimans, 2026) tests cross-application business workflows, and DeepPlanning (Zhang et al., 2026a) evaluates planning under verifiable constraints. VitaBench (He et al., 2025) evaluates multi-turn service interaction and cross-scenario coordination, while VitaBench 2.0 (Chen et al., 2026) tests personalized and proactive assistance over long-term interactions. Agents' Last Exam (ALE) (Sun et al., 2026) evaluates long-horizon professional tasks with verifiable outcomes. We use the 34-task pure-text subset of WildClawBench and version 1.0.6 of AutomationBench. For the agent harnesses,

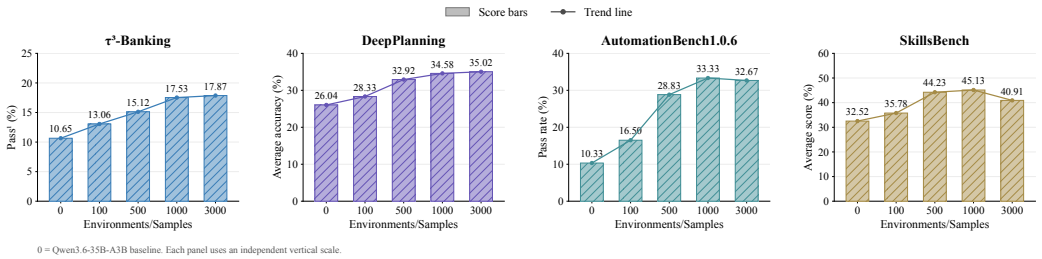


Figure 3: Effect of SFT training-set size on Qwen3.6-35B-A3B across four benchmarks. The zero-sample setting denotes the initial backbone.

we use OpenHands for SkillsBench with skills enabled, Claude Code for ALE, and OpenClaw for WildClawBench. We report the average score or reward for WildClawBench and both VitaBench versions, Avg@3 for SkillsBench, the pass rate for τ^3-Banking, AutomationBench, and ALE, and the average accuracy for DeepPlanning.

4.2 Main Results

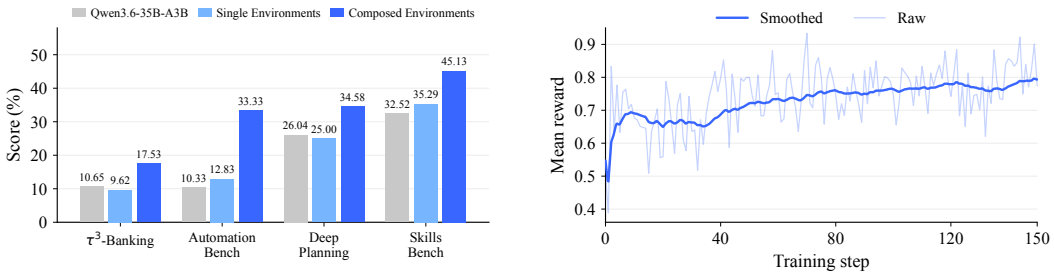
CompoWorld improves on all eight benchmarks in Table 1, averaging a 9.17-point gain over Qwen3.6-35B-A3B. AutomationBench task success rises from 10.33% to 32.33% (+22.00 points), more than tripling the backbone's pass rate and exceeding GPT-5.4, Gemini-3.1 Pro, and GLM-5.2, while approaching DeepSeek-V4-Flash (36.33%). CompoWorld also leads all six compared agent-specialized 35B-A3B models on this benchmark, outperforming the strongest reported baseline, Occamy-1.0, by 4.73 points. SkillsBench and VitaBench improve by 15.19 and 10.50 points, respectively; the smaller gain on VitaBench 2.0 (+1.62) suggests more limited transfer to personalized, long-term assistance. Under our evaluation protocol, CompoWorld exceeds Apodex 1.1 Mini on all six shared benchmarks. Its advantage is strongest in workflow execution: frontier models still lead on τ^3-Banking, DeepPlanning, and VitaBench 2.0. Appendix E reports gains across all six AutomationBench domains, with mean partial-credit scores increasing from 41.94 to 72.68. These gains complement the improvement in complete-task success, although the 32.33% pass rate shows that satisfying every requirement remains challenging.

4.3 Analysis

4.3.1 Effect of Environment Scaling

We examine how agent performance changes as the number of training environments increases. Since prior work uses different environment granularities, we follow the definition in Section 2: each service set C defines a compositional environment $E_C = \bigotimes_{E_i \in C} E_i$. The scaling unit is therefore a composed environment, while individual services are reusable building blocks. For a library of n services and compositions of size K, the number of possible service sets is $N_K = \binom{n}{K} = \frac{n!}{K!(n-K)!}$. For fixed K and increasing n, $N_K \sim n^K/K!$, so the composition space grows rapidly even with a limited service library. With $n = 448$, composing 5 services yields approximately 1.47×10^{11} possible sets, while composing 7 yields approximately 6.86×10^{14}. These counts describe candidate compositions; task synthesis and verification determine which support meaningful, solvable workflows. This combinatorial space allows us to expand the training environment set by recombining existing services, without implementing a new service for every additional example. When each verified trajectory comes from a distinct composition, adding an SFT example also adds a training environment. Environment scaling can thus be realized through data scaling, with reusable services supplying the executable foundation for each new sample. We focus this experiment on SFT, considering training sets of 100, 500, 1,000, and 3,000 examples, each paired with a distinct compositional environment. The comparison uses the same backbone, service library, and SFT recipe across scales, with no RL stage.

Figure 3 shows that all four benchmarks improve over the backbone with as few as 100 SFT examples. At 3K, τ^3-Banking rises from 10.65 to 17.87 and DeepPlanning from 26.04 to 35.02.



(a) Single vs. Composed Environments (1K SFT)

(b) Reinforcement learning dynamics

Figure 4: **Environment composition and RL training.** (a) Comparison between single-environment scaling and composed-environment scaling; (b) Mean trajectory reward during the RL stage.

AutomationBench reaches 33.33% at 1K and 32.67% at 3K, while SkillsBench reaches 44.23 at 500, peaks at 45.13 at 1K, and declines to 40.91 at 3K, still above the backbone's 32.52. Thus, service quality and composability are central to future environment scaling. Adding a service to a library of n services creates $\binom{n}{K-1}$ additional candidate compositions of size K, so expanding the library complements generating more compositions from the existing pool. The practical challenge is to turn this potential into training coverage by selecting compatible services, constructing substantive cross-service dependencies, and verifying the resulting tasks. A growing collection of reliable, reusable services can support continued data generation and make workflow diversity an explicit scaling dimension. Overall, rather than treating environment scaling as independent of sample scaling, we view reusable service composition as the mechanism for scaling the diversity and structural complexity of training data.

4.3.2 Single vs. Composed Environment Scaling

The scaling experiment above varies both the amount of training data and the coverage of environments. To determine whether environment composition itself provides benefits over training on isolated environments, we compare the two settings under a fixed data budget and the same number of environments: one model is trained on 1K trajectories from single environments, while the other is trained on 1K trajectories from composed environments using the same SFT procedure. Figure 4(a) shows that composed-environment training consistently yields substantial performance improvements over single-environment training across all four completed comparisons. For example, on τ^3-Banking, the score increases from 9.62 to 17.53 (+7.91 points), while on AutomationBench, it rises from 12.83 to 33.33 (+20.50 points). We further observe that training on single environments does not consistently improve performance over the base model. In particular, on both τ^3-Banking and DeepPlanning, single-environment SFT leads to a slight performance degradation relative to the base Qwen3.6-35B-A3B model. These results suggest that composed environments provide a more effective training signal by introducing richer interaction structures, greater task complexity, and broader data diversity, which in turn can promote stronger generalization.

4.3.3 RL Training Dynamics

In our experiments, after SFT, uniform rubric rewards are already high because most of the easy criteria have been satisfied, while full-task completion remains difficult. Filtering for lower-scoring tasks does not resolve this difficulty, because the model can struggle to satisfy even a single criterion in difficult cross-service tasks. This phenomenon is especially pronounced for stronger models; many benchmarks indicate that models can achieve acceptable rubric scores but have relatively low pass rates. Our experiments find that when scores are already high, the RL reward is difficult to increase further. Therefore, completion-focused weighting is designed specifically to allocate the learning signal to these remaining bottlenecks. Figure 4(b) shows the mean trajectory reward during RL and its smoothed trend. The smoothed reward reaches approximately 0.69 within the first ten steps, settles around 0.65–0.67, and then resumes its increase. It reaches about 0.75 at step 80 and ends near 0.79 at step 150. This pattern suggests an initial adjustment period followed by sustained progress on the training objective. Appendix C presents the separate evaluation-set comparison with and

without reweighting, together with a gradient interpretation of the reward. The results show that the completion-focused rubric reward can yield more stable gains during RL training, with RL improving on SFT by 1.2 points on average across eight benchmarks. At the same time, we also find that RL does not always bring performance gains: although it improves performance substantially on some benchmarks, such as SkillsBench, performance also declines slightly on τ^3-Banking. Overall, our performance improvements are primarily driven by SFT.

5 Related Work

5.1 Environment Scaling for LLM Agents

Automated environment construction supports interactive agent training while reducing reliance on costly or restricted real services (Li et al., 2026a; Fang et al., 2025). DreamGym simulates transitions and feedback through a reasoning-based experience model (Chen et al., 2025). Executable methods synthesize environments and verifiable tasks (Cai et al., 2025; Wang et al., 2026); EnvScaler separates environment skeleton construction from scenario generation and rule-based validation (Song et al., 2026). Other work uses dependency-graph expansion and topology-aware trajectory synthesis (Tu et al., 2026; Xu et al., 2026), generates interactive websites (Wu et al., 2026b; Zhang et al., 2026b), or configures existing software with realistic data (Aggarwal et al., 2026). Learner-adaptive transformations and ability-aware curricula further emphasize training utility beyond environment count (Huang et al., 2026; Zhu et al., 2026). Closely related, Terminal-Universe reconstructs workspaces from agent trajectories, synthesizes tasks in which changes to a writable codebase depend on evidence from a related read-only codebase, and extends tasks across persistent workspace states (Wu et al., 2026a). CompoWorld instead reuses independently executable, stateful services, composing them through task-specific causal dependency graphs. Typed service states and namespaced tool interfaces let the same implementations support multiple application workflows, making service combinations and cross-service requirements explicit, controllable dimensions of training-environment generation.

5.2 Compositional Generalization for LLM Agents

For LLM agents, compositional generalization requires reusing familiar tools and skills in new task structures while respecting dependencies between actions and environment states. CompWoB (Furuta et al., 2023) directly studies this challenge by composing basic web tasks, showing that strong performance on individual tasks does not reliably transfer to their combinations. AppWorld (Trivedi et al., 2024) extends evaluation to workflows spanning multiple applications, where agents must coordinate API calls and state changes to achieve user goals. On the method side, Voyager builds a library of reusable executable skills for solving new tasks (Wang et al., 2023), while Compositional Skill Routing decomposes requests, retrieves relevant MCP skills, and assembles dependency-aware plans (Gao, 2026). These studies motivate both evaluating task composition and equipping agents with reusable capabilities. CompoWorld pursues a complementary direction through the training distribution: it recombines independently executable services and their causal dependencies to generate verified cross-environment tasks for SFT and RL. The goal is to develop agents that transfer their knowledge of individual services to new workflows by learning how information and state changes connect across services.

6 Conclusion

We presented CompoWorld, a framework for compositional environment scaling that builds reusable executable services and connects them through task-specific causal dependencies. Its automated construction and verification pipeline generates cross-environment tasks for SFT and RL, while the Completion-Focused Rubric Reward emphasizes unmet requirements to encourage full task completion. Experiments across eight agent benchmarks demonstrate the benefits of training on these composed workflows. On AutomationBench, it surpasses frontier models such as Claude Opus 4.6 and leads all compared agent-specialized 35B-A3B models. These results highlight service composition as a promising dimension of environment scaling that can increase the complexity and diversity of training data and help LLM agents coordinate familiar capabilities in new workflows.

AI USE STATEMENT

Large language models are used as components of the CompoWorld framework for environment and task synthesis, trajectory generation, selective world-model simulation, and evaluation, as described in the main text and appendix. Beyond these methodological uses, generative AI assisted with English translation, language polishing, figure and table presentation, and bibliographic checks against primary sources. The authors retain full responsibility for the technical content, experimental results, citations, conclusions, and the accuracy and integrity of the final manuscript.

REFERENCES

Accio Team. Occamy-1.0: Open Pareto-frontier 35B Intelligence for Co-work. *arXiv preprint arXiv:2609.11977*, 2026.

Pranjal Aggarwal, Graham Neubig, and Sean Welleck. Gym-Anything: Turn any Software into an Agent Environment. *arXiv preprint arXiv:2604.06126*, 2026.

Anthropic. Claude Opus 4.6 System Card. System card, 2026.

Apodex Team. Apodex 1.1: Scaling Agentic Intelligence for Complex Work. *arXiv preprint arXiv:2608.23283*, 2026.

Lei Bai, Zongsheng Cao, Yang Chen, Zhiyao Cui, Shangheng Du, Yue Fan, Shiyang Feng, Zijie Guo, Haonan He, Liang He, Xiaohan He, Shuyue Hu, Yusong Hu, Songtao Huang, Yichen Jiang, Hao Li, Xin Li, Dahua Lin, Weihao Lin, Fenghua Ling, Dongrui Liu, Zhuo Liu, Wenjie Lou, Runmin Ma, Chunjiang Mu, Haoyang Peng, Tianshuo Peng, Jinxin Shi, Luohe Shi, Boyuan Sun, Zelin Tan, Shengji Tang, Yan Teng, Qianyi Wang, Xiaosong Wang, Yiming Wu, Yi Xie, Xiangchao Yan, Jingqi Ye, Peng Ye, Fangchen Yu, Jiakang Yuan, Bihao Zhan, Bo Zhang, Chen Zhang, Shufei Zhang, Shuaiyu Zhang, Wenlong Zhang, Yiqun Zhang, Junpeng Zhao, Zhijie Zhong, Bowen Zhou, and Yuhao Zhou. Scaling the Horizon, Not the Parameters: Reaching Trillion-Parameter Performance with a 35B Agent. *arXiv preprint arXiv:2606.30616*, 2026.

Shihao Cai, Runnan Fang, Jialong Wu, Baixuan Li, Xinyu Wang, Yong Jiang, Liangcai Su, Liwen Zhang, Wenbiao Yin, Zhen Zhang, Fuli Feng, Pengjun Xie, and Xiaobin Wang. AutoForge: Automated Environment Synthesis for Agentic Reinforcement Learning. *arXiv preprint arXiv:2512.22857*, 2025.

Yuxin Chen, Yi Zhang, Zhengzhou Cai, Yaorui Shi, Zhiyuan Yao, Chenhang Cui, Jingnan Zheng, Yaqi Huo, Xi Su, Qi Gu, Xunliang Cai, Xiang Wang, An Zhang, and Tat-Seng Chua. VitaBench 2.0: Evaluating Personalized and Proactive Agents in Long-Term User Interactions. *arXiv preprint arXiv:2605.27141*, 2026.

Zhaorun Chen, Zhuokai Zhao, Kai Zhang, Bo Liu, Qi Qi, Yifan Wu, Tarun Kalluri, Sara Cao, Yuanhao Xiong, Haibo Tong, Huaxiu Yao, Hengduo Li, Jiacheng Zhu, Xian Li, Dawn Song, Bo Li, Jason Weston, and Dat Huynh. Scaling Agent Learning via Experience Synthesis. *arXiv preprint arXiv:2511.03773*, 2025.

DeepSeek-AI. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Shuangrui Ding, Xuanlang Dai, Long Xing, Shengyuan Ding, Ziyu Liu, Yang JingYi, Penghui Yang, Zhixiong Zhang, Xilin Wei, Xinyu Fang, Yubo Ma, Haodong Duan, Jing Shao, Jiaqi Wang, Dahua Lin, Kai Chen, and Yuhang Zang. WildClawBench: A Benchmark for Real-World, Long-Horizon Agent Evaluation. *arXiv preprint arXiv:2605.10912*, 2026.

Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, Longxiang Liu, Shijue Huang, Zhenyu Li, Yang Zhao, Xiaoshuai Song, Xiaoxi Li, Jiajie Jin, Yutao Zhu, Hanbin Wang, Fangyu Lei, Qinyu Luo, Mingyang Chen, Zehui Chen, Jiazhan Feng, Ji-Rong Wen, and Zhicheng Dou. AgentWorld: Scaling Real-World Environment Synthesis for Evolving General Agent Intelligence. *arXiv preprint arXiv:2604.18292*, 2026.

Earendil Works. Pi Agent Harness. GitHub repository, 2026.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, Shibin Wu, Zhengwei Tao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Towards General Agentic Intelligence via Environment Scaling. *arXiv preprint arXiv:2509.13311*, 2025.

Hiroki Furuta, Yutaka Matsuo, Aleksandra Faust, and Izzeddin Gur. Exposing Limitations of Language Model Agents in Sequential-Task Compositions on the Web. *arXiv preprint arXiv:2311.18751*, 2023.

Xueping Gao. Compositional Skill Routing for LLM Agents: Decompose, Retrieve, and Compose. *arXiv preprint arXiv:2606.18051*, 2026.

GLM-5 Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. GLM-5: from Vibe Coding to Agentic Engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Google DeepMind. Gemini 3.1 Pro Model Card. Model card, 2026.

Wei He, Yueqing Sun, Hongyan Hao, Xueyuan Hao, Zhikang Xia, Qi Gu, Chengcheng Han, Dengchang Zhao, Hui Su, Kefeng Zhang, Man Gao, Xi Su, Xiaodong Cai, Xunliang Cai, Yu Yang, and Yunke Zhao. VitaBench: Benchmarking LLM Agents with Versatile Interactive Tasks in Real-world Applications. *arXiv preprint arXiv:2509.26490*, 2025.

Chengsong Huang, Zifeng Wang, Rujun Han, Jun Yan, Yanfei Chen, Zoey CuiZhu, Ke Jiang, Peng Xia, Han Yu, Yufan Zhuang, Yifei Ming, Jiaqi Pan, Bhavana Dalvi Mishra, Jiaxin Huang, Burak Gokturk, Tomas Pfister, and Chen-Yu Lee. EnvHarness: Awakening Static Worlds for Agent Learning. *arXiv preprint arXiv:2608.19880*, 2026.

Jiachun Li, Zhuoran Jin, Tianyi Men, Yupu Hao, Kejian Zhu, Lingshuai Wang, Dongqi Huang, Longxiang Wang, Shengjia Hua, Lu Wang, Jinshan Gao, Hongbang Yuan, Ruilin Xu, Kang Liu, and Jun Zhao. Agentic Environment Engineering for Large Language Models: A Survey of Environment Modeling, Synthesis, Evaluation, and Application. *arXiv preprint arXiv:2606.12191*, 2026a.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, Chujun Tao, Binxu Li, Xuandong Zhao, Hejia Geng, Xiaojun Wu, Junwei Zhou, Xiaokun Chen, Hanwen Xing, Yubo Li, Qunhong Zeng, Di Wang,

Yuanli Wang, Roey Ben Chaim, Penghao Jiang, Haotian Shen, Luyang Kong, Xinyi Liu, Runhui Wang, Xuanqing Liu, Jiachen Li, Xin Lan, Yueqian Lin, Wengao Ye, Junwei He, Songlin Li, Yue Zhang, Yipeng Gao, Yijiang Li, Ze Ma, Liqiang Jing, Tianyu Wang, Kaixin Li, Yiqi Xue, Haoran Lyu, Yizhuo He, Yuchen Tian, Shutong Wu, Bowei Wang, Yixuan Gao, Bo Chen, Litong Liu, Sikai Cheng, Jiajun Bao, Shuaicheng Tong, Shuwen Xu, Terry Yue Zhuo, Tinghan Ye, Qi Qi, Miao Li, Longtai Liao, Zelin Tan, Chang Shi, Xilin Tang, Srinath Tankasala, Boqin Yuan, Yaoyao Qian, Jianhong Tu, Chenguang Wang, Yizhou Sun, Wei Wang, Aaron Taylor, Ziyue Yang, Changkun Guan, Zhikang Dong, Xinyu Zhang, Steven Dillmann, Han chung Lee, and Dawn Song. SkillsBench: Benchmarking How Well Agent Skills Work Across Diverse Tasks. *arXiv preprint arXiv:2602.12670*, 2026b.

Kate McCurdy, Paul Soulos, Paul Smolensky, Roland Fernandez, and Jianfeng Gao. Toward Compositional Behavior in Neural Models: A Survey of Current Views. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 9323–9339. Association for Computational Linguistics, 2024.

Moonshot AI. Kimi K2.6: Advancing Open-Source Coding. Technical blog, 2026.

Nex AGI. Nex-N2-mini. Official model card, 2026.

OpenAI. GPT-5.4 Thinking System Card. System card, 2026.

Ornith. Ornith-1.5: From Self-Scaffolding to Self-Improvement. Official release article, 2026.

Qwen Team. Qwen3.5: Towards Native Multimodal Agents. Technical blog, February 2026a.

Qwen Team. Qwen3.6-35B-A3B: Agentic Coding Power, Now Open to All. Technical blog, April 2026b.

Qwen Team. Qwen3.8-Max: A New Bar for Coding and Cowork. Technical blog, August 2026c.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Daniel Shepard and Robin Salimans. AutomationBench. *arXiv preprint arXiv:2604.18934*, 2026.

Quan Shi, Alexandra Zytek, Pedram Razavi, Karthik Narasimhan, and Victor Barres. τ-Knowledge: Evaluating Conversational Agents over Unstructured Knowledge. *arXiv preprint arXiv:2603.04370*, 2026.

Xiaoshuai Song, Haofei Chang, Guanting Dong, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. EnvScaler: Scaling Tool-Interactive Environments for LLM Agent via Programmatic Synthesis. In *Findings of the Association for Computational Linguistics: ACL*, pp. 8326–8357. Association for Computational Linguistics, 2026.

Yiyou Sun, Xinyang Han, Weichen Zhang, Yuanbo Pang, Tianyu Wang, Yuhan Cao, Yixiao Huang, Chris Duroiu, Haoyun Zhang, Jeffrey Lin, Weishu Zhang, Tyler Zeng, Ying Yan, Bo Liu, Hanson Wen, Mingyang Xu, Xiaoyuan Liu, Zimeng Chen, Weiyan Shi, Amanda Dsouza, Vincent Sunn Chen, Patrick Bryant, Carl Boettiger, Yamini Rangan, Bradley Rothenberg, Kyle Steinfeld, Arvind Rao, Tapio Schneider, Georgios Yannakakis, Laure Zanna, Kaan Ozbay, Ida Sim, Tarek Zohdi, George Em Karniadakis, Jack Gallant, Teresa Head-Gordon, Yushan Li, Wenxi Deng, Tao Sun, Huiqi Wang, Zhun Wang, Justin Xu, Chris Yuhao Liu, Yafei Cheng, Rongwang Hu, Aras Bacho, Shengcao Cao, Zengyi Qin, Yixiong Chen, Hengduan Fan, Hao Liu, Lin Zeng, Shashank Muralidhar Bharadwaj, Litian Gong, Yingxuan Yang, Maojia Song, Ruheng Wang, Zongzheng Zhang, Honglin Bao, Shuo Lu, Jianhong Tu, Zhonghua Wang, Zheng Zhang, Zijiao Chen, Yanqiong Jiang, Zhendong Li, Bohan Lyu, Chang Ma, Peiran Xu, Benran Zhang, Shangding Gu, Haoyue Hua, Haoyang Li, Wanzhe Liao, Chengzhi Liu, Junbo Peng, Haoran Sun, Zechen Xu, Bo Chen,

Jiayi Cheng, Yi Jiang, Keying Kuang, Yuan Li, Youbang Pan, Ziyan Rao, Alexander Schubert, Yifan Shen, Vincent Siu, Xiatao Sun, Kangqi Zhang, Xiaopan Zhang, Yuchen Zhu, Ishaan Singh Chandok, Lei Ding, Jingxuan Fan, Andrew Glover, Jiaming Hu, Yiran Hu, Wenbo Huang, Zixin Jiang, Haoran Jin, Lukas Kim, Ming Liu, Yang Liu, Alireza Rafiei, Xuhuan Shen, Kunyang Sun, Sophia Sun, Ting Sun, Eric Wang, Yixin Wang, Hanwen Xing, Sihan Xu, Yuzheng Xu, Zhongxing Xu, Zhiling Yan, Boqin Yuan, Ruiqi Zhang, Yifan Zhang, Zibo Zhao, Liana, Santanu Bosu Antu, Haoyue Bai, Carlo Bosio, Joseph Cavanagh, Patricia Cavazos-Rehg, Tianxing Chen, Xuewen Chen, Yipu Chen, Chenyu Zhu, Chen Dai, Stefano De Castro, Yunfu Deng, Kaustubh Dhole, Jiayuan Ding, Chenchen Du, Zhehang Du, Hao Fan, Run-Ze Fan, Hengyu Fu, Shi Gu, Yifan Gu, Charlie Guo, Baihe Huang, Baixiang Huang, Rimika Jaiswal, Zhihan Jiang, Ran Jin, Erin Kasson, Xin Lan, Joseph Lee, Deren Lei, Chenyu Li, Daofeng Li, Haitao Li, Hongwei Li, Jingyan Li, Xiao Li, Yi Li, Yinsheng Li, Yuangang Li, Zhixu Li, Wenyu Liang, Longtai Liao, Kevin Qinghong Lin, Andy Zeyi Liu, Che Liu, Jiaming Liu, Kaiyuan Liu, Xuan Liu, Pan Lu, Wenbo Lv, Yicheng Lyu, Qiuyang Mang, Kyle Montgomery, Yuzhou Nie, Ruoxi Ning, Jorin Overwiening, Xu Pan, Layna Paraboschi, Core Francisco Park, Justin Purnomo, Swati Rajwal, Scott Rankin, Bixuan Ren, Yiren Rong, HaoYang Shang, Ventus Shaw, Fiona Shen, Jiawei Shen, Minqi Shi, Shi Qiu, Huaxiu Yao, Tianneng Shi, Jonah So, Vladislav Susoy, Hannah Szlyk, Haocheng Wang, Jialu Wang, Wei Wang, Xinyu Wang, Zehao Wang, Dowling Wong, Angela Wu, Dehao Wu, Fangyu Wu, Mengyuan "Millie" Wu, Yu Wu, Yuchen Wu, Yuhao Wu, Qingpo Wuwu, Weihang Xiao, Yongyi Xiong, Fan Xu, Ruiling Xu, Mingxuan Yan, Benjamin Yang, Jirong Yang, Sen Yang, Xiaoli Yang, Yushi Yang, Haoran Ye, Xiaohu Yu, Zhengming Yu, Chenlong Zhang, Chi Zhang, Hanning Zhang, Hanwen Zhang, Junge Zhang, Kunpeng Zhang, Song Zhang, Wenjin Zhang, Wenshuo Zhang, Ying Zhang, Yizhi Zhang, Brian Zhao, Qijian Zhao, Yimin Zhao, Yuhaohua Zheng, Liwei Zhou, Tianyue Zhou, Sichen Zhu, Siqi Zhu, Yan Zhu, Yishu Zhu, Jierui Zuo, Chonghao Cai, Helena Casademunt, Wenjia Chen, Cheng Cheng, Nawen Deng, Rao Fu, Tianfu Fu, Yifan Han, He Ren, Zhenyu He, Qiao Jin, Langlang Li, Yuetai Li, Sylvia Liu, Lu Lu, Luqing Zhou, Subhabrata Mukherjee, Yunqi Ouyang, Yin Ren, Dawei Shi, Haoran Wu, Zhiyue Wu, Hannah Yao, Zhuoran Yi, Jenny Yu, Rhea Zhan, Hang Zhou, Blake Zhu, Junfan Zhu, Alan Yuille, Yang Liu, Russell Alan Poldrack, Jiachen Li, Zhenglu Li, Molei Tao, Jing Huang, Wenqi Shi, Costas Spanos, Lichao Sun, Chenguang Wang, Orson Xu, Zhen Dong, Hector Gomez, Aylin Caliskan, Ali Emami, Haimin Hu, Zhi Li, Lihui Liu, Murphy Niu, Yi Shao, Jianxin Sun, Mikko Tolonen, Ting Wang, Sanjiv Das, Yanjun Gao, Wenbo Guo, Erika J Schneider, Zhiyong Lu, Yian Ma, Mark Mueller, Radha Poovendran, Somayeh Sojoudi, Yinglun Zhu, and Dawn Song. Agents' Last Exam. *arXiv preprint arXiv:2606.05405*, 2026.

The BigBang Team. BigBang: Pursuing Open-Ended Intelligence through Self-Evolving Synthesis of Verifiable Frontier Tasks. Technical report, 2026.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076. Association for Computational Linguistics, 2024.

Dunwei Tu, Hongyan Hao, Hansi Yang, Yihao Chen, Yi-Kai Zhang, Zhikang Xia, Yu Yang, Yueqing Sun, Xingchen Liu, Furao Shen, Qi Gu, Hui Su, and Xunliang Cai. ScaleEnv: Scaling Environment Synthesis from Scratch for Generalist Interactive Tool-Use Agent Training. *arXiv preprint arXiv:2602.06820*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*, 2023.

Zhaoyang Wang, Canwen Xu, Boyi Liu, Yite Wang, Siwei Han, Zhewei Yao, Huaxiu Yao, and Yuxiong He. Agent World Model: Infinity Synthetic Environments for Agentic Reinforcement Learning. *arXiv preprint arXiv:2602.10090*, 2026. Accepted to ICML 2026.

Jie Wu, Zhenru Zhang, Beichen Zhang, Xuwu Wang, Yuhui Su, Mouxiang Chen, Peng Wang, Zhihai Wang, Que Shen, Hao Zhou, An Yang, Fei Huang, Yujiu Yang, and Dayiheng Liu. Terminal-Universe: Turning Agent Trajectories into Scalable Terminal Environments. *arXiv preprint arXiv:2609.04148*, 2026a.

Yifan Wu, Yiran Peng, Yiyu Chen, Jianhao Ruan, Zijie Zhuang, Cheng Yang, Jiayi Zhang, Man Chen, Yenchi Tseng, Zhaoyang Yu, Liang Chen, Yuyao Zhai, Bang Liu, Chenglin Wu, and Yuyu Luo. AutoWebWorld: Synthesizing Infinite Verifiable Web Environments via Finite State Machines. *arXiv preprint arXiv:2602.14296*, 2026b.

Minrui Xu, Zilin Wang, Mengyi Deng, Zhiwei Li, Zhicheng Yang, Xiao Zhu, Yinhong Liu, Boyu Zhu, Baiyu Huang, Chao Chen, Heyuan Deng, Fei Mi, Lifeng Shang, Xingshan Zeng, and Zhijiang Guo. EnvFactory: Scaling Tool-Use Agents via Executable Environments Synthesis and Robust RL. *arXiv preprint arXiv:2605.18703*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023.

Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, Chenxu Lv, and Junyang Lin. DeepPlanning: Benchmarking Long-Horizon Agentic Planning with Verifiable Constraints. *arXiv preprint arXiv:2601.18137*, 2026a.

Ziyun Zhang, Zezhou Wang, Xiaoyi Zhang, Zongyu Guo, Jiahao Li, Bin Li, and Yan Lu. InfiniteWeb: Scalable Web Environment Synthesis for GUI Agent Training. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28465–28492. Association for Computational Linguistics, 2026b.

Kejian Zhu, Zhuoran Jin, Dongqi Huang, Hongbang Yuan, Yupu Hao, Kang Liu, and Jun Zhao. Beyond Simply Environment Scaling: Designing Effective Environment Distributions for Multimodal Agent Learning. *arXiv preprint arXiv:2608.03571*, 2026.

Yuxin Zuo, Zikai Xiao, Li Sheng, Fei Huang, Jianhong Tu, Yuxuan Liu, Tianyi Tang, Xiaomeng Hu, Yang Su, Qingfeng Lan, Yantao Liu, Qin Zhu, Yinger Zhang, Bowen Yu, Haiquan Zhao, Haiyang Xu, Jianxin Yang, Jiayang Cheng, Junyang Wang, Lianghao Deng, Mingfeng Xue, Tianyi Bai, Yang Fan, Yubo Ma, Yucheng Li, Zeyu Cui, Zhihai Wang, Zhihui Xie, Zhuorui Ye, An Yang, Dayiheng Liu, Jingren Zhou, and Ning Ding. Qwen-AgentWorld: Language World Models for General Agents. *arXiv preprint arXiv:2606.24597*, 2026.

A Corpus and Taxonomy

Each service is packaged as an independently executable environment E_i, with its own persistent state and namespaced tools. Because all services follow the same state and execution conventions, they can be composed directly using the product construction $E_C = \bigotimes_{E_i \in C} E_i$. The pipeline produces 448 services spanning 10,130 tools. To characterize their breadth, we assign each service to a single application domain. Figure 5 shows the resulting distribution. The corpus is deliberately broad, not concentrated. No single domain accounts for more than a quarter of the services, and thirteen domains each contribute a notable share. The software development and DevOps domain accounts for the largest share at 22%, reflecting the abundance of publicly specified developer tooling. This is followed by a broad range of business and consumer software, including productivity and collaboration at 12%, data and analytics, and other domains. This spread makes the corpus a useful training substrate. When measured by tool count instead of service count, the ordering is broadly similar, but business application domains rise. Marketing, sales, and CRM, along with data and analytics, contribute disproportionately many tools because their services tend to expose larger APIs. The 10,130 tools are therefore spread even more evenly across domains than the services are.

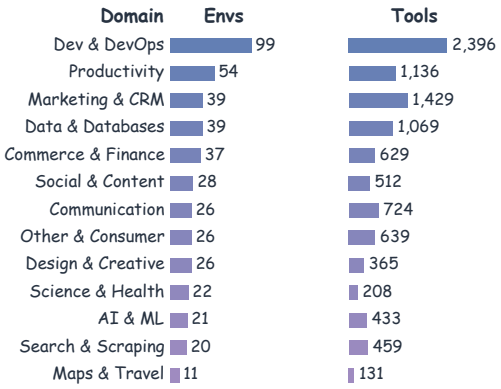


Figure 5: Domain counts of 448 environments and 10,130 tools, with shared row colors.

B Policy Optimization

We use Group Relative Policy Optimization (GRPO) (Shao et al., 2024). The reward in Eq. 5 is normalized within each rollout group to obtain the advantage:

$$\widehat{A}_{i,t} = \frac{R_i - \overline{R}}{\sigma_R + \delta}, \qquad \overline{R} = \frac{1}{N}\sum_{i=1}^{N} R_i, \qquad \sigma_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_i - \overline{R})^2}, \tag{6}$$

where $\delta > 0$ ensures numerical stability. The same outcome advantage is assigned to all agent-generated tokens in a trajectory. Let $y_{i,t}$ be the t-th such token and $c_{i,t}$ its full context, including the instruction, previous agent tokens, and available tool observations. The token-level probability ratio is

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid c_{i,t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid c_{i,t})}. \tag{7}$$

We maximize

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{\rho_i\}_{i=1}^{N}\sim\pi_{\theta_{\text{old}}}(\cdot|x;E_C)} \Bigg[\frac{1}{N}\sum_{i=1}^{N} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \Big\{ \min\Big(r_{i,t}(\theta)\widehat{A}_{i,t}, \\ \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\, \widehat{A}_{i,t} \Big) - \beta D_{\text{KL}}(\pi_\theta(\cdot \mid c_{i,t}) \,\|\, \pi_{\text{ref}}(\cdot \mid c_{i,t})) \Big\} \Bigg], \tag{8}$$

where $\mathcal{D}$ is the training task set, ϵ is the clipping threshold, β controls the KL penalty, and π_{ref} is the fixed reference policy. Here $|y_i|$ counts only agent-generated tokens. Tool observations enter the context but are excluded from the loss. Thus, the policy update follows standard GRPO, while the completion-focused reward determines which outcomes receive higher relative advantages.

C Analysis of Completion-Focused Rubric Reward

The completion-focused weighting allocates larger gradient coefficients to rubric terms with larger completion gaps. To see this directly in GRPO, fix a rollout group and write $W = \sum_j w_j$. Subtracting

the group mean reward gives

$$\widehat{A}_i = \frac{\sum_j w_j(b_{ij} - p_j)}{W(\sigma_R + \delta)}. \tag{9}$$

Let $z_i = |y_i|^{-1} \sum_t \nabla_\theta \log \pi_\theta(y_{i,t} \mid c_{i,t})|_{\theta=\theta_{\text{old}}}$, and define $g_j = N^{-1} \sum_i (b_{ij} - p_j) z_i$ as the empirical gradient signal for rubric term j. At the update origin, where the probability ratio is one and clipping is inactive, the policy-reward part of Eq. 8 has gradient

$$\nabla_\theta \widehat{J}_{\text{policy}}\Big|_{\theta_{\text{old}}} = \frac{1}{\sigma_R + \delta} \sum_{j=1}^{M} \alpha_j g_j, \qquad \alpha_j = \frac{\lambda + (1 - p_j)}{\sum_k [\lambda + (1 - p_k)]}. \tag{10}$$

Thus, $1 - p_j$ is precisely the completion gap controlling term j's gradient weight. The reward normalization and group standard deviation are shared across terms, so

$$p_j < p_k \quad \Longrightarrow \quad \frac{\alpha_j}{\alpha_k} = \frac{\lambda + 1 - p_j}{\lambda + 1 - p_k} > 1. \tag{11}$$

Unlike uniform rubric averaging, which assigns equal coefficients to all g_j, this update gives greater relative weight to requirements the policy has yet to satisfy reliably. As a requirement becomes more consistently completed, its relative weight decreases compared with terms whose pass rates remain unchanged, shifting emphasis toward the remaining completion gaps. The positive floor λ retains every requirement. This establishes adaptive weighting of gradient terms; their norms need not follow the same ordering because the signal g_j also depends on sampled outcomes and is zero when every rollout fails that term. The derivation concerns the policy-reward gradient at the update origin.

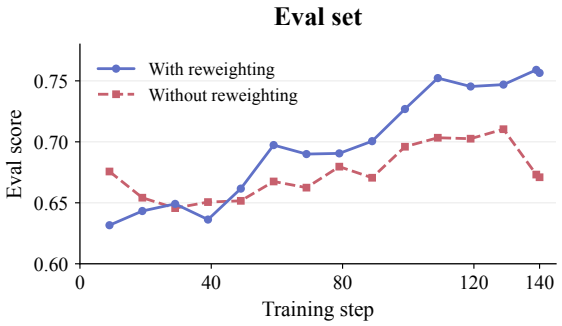


Figure 6: Smoothed evaluation-set scores with and without reward reweighting.

Figure 6 compares evaluation-set scores with and without reweighting. The reweighted run starts lower (approximately 0.63 versus 0.68), but moves ahead around step 50 and remains ahead thereafter. Near step 110, its smoothed score reaches about 0.75, versus 0.70 without reweighting. At step 140, the scores are approximately 0.76 and 0.67, a gap of about 0.09. This advantage persists across multiple evaluations. The pattern supports the benefit of reweighting in these runs and is consistent with the gradient allocation above: less frequently satisfied criteria receive greater relative emphasis.

D Implementation Details

We use GLM-5.3 for environment and task synthesis, which requires strong agentic capabilities, and DeepSeek-V4-Flash (0731) to generate trajectories. By default, each task composes $K = 5$ services, with five additional distractor services for tool discovery. The walk length L is sampled randomly from 7 to 12. The agent uses four meta-tools: `list_services` discovers available services, `list_tools` lists their tools, `describe_tool` retrieves argument schemas, and `call_tool` executes a selected tool. Appendix G illustrates a calendar environment, and Appendix H gives a complete tool-use trajectory. We train Qwen3.6-35B-A3B on 3K successful trajectories for SFT, followed by 1K tasks for RL. Tables 1 and 2 report the resulting post-RL checkpoint. SFT runs for three epochs with a global batch size of 32 and a learning rate of 10^{-4}. RL uses GRPO with Adam at a learning rate of 10^{-5}, lower and upper clipping parameters $(0.20, 0.28)$, $N = 8$, $\lambda = 0.01$, and zero KL and entropy coefficients.

E Domain-Level Results on AutomationBench

Table 2 reports partial-credit scores on AutomationBench 1.0.6. CompoWorld improves in every domain, raising the mean from 41.94 to 72.68. HR shows the largest gain (+49.40 points), followed by Marketing (+32.51) and Sales (+31.45); Finance, Support, and Operations improve by 21.49–26.56 points. CompoWorld exceeds both GPT-5.4 and GLM-5.2 in five of six domains, with Support as the exception. It ranks first among the listed models in HR and second in Finance and Operations. The gains therefore extend across business functions, rather than being driven by a single domain. Tables 1 and 2 capture complementary aspects of performance. The domain scores award partial credit for satisfying task requirements, whereas the overall pass rate measures complete success. The increase in both metrics indicates progress toward executing whole workflows. However, the 32.33% pass rate shows that completing every requirement remains difficult, even when substantial partial progress is made. Sales remains the lowest-scoring domain at 58.93, despite its large improvement, suggesting room to strengthen the dependencies and constraints involved in these workflows.

Table 2: Domain-level results on AutomationBench 1.0.6 (average score, %). Best scores are **bold**; second-best scores are underlined. Green values show score-point gains over Qwen3.6-35B-A3B.

Model	Finance	HR	Marketing	Operations	Sales	Support
GPT-5.4	67.39	62.05	71.50	77.21	53.56	78.05
Gemini-3.1 Pro	**75.26**	72.07	78.37	72.26	61.37	74.05
DeepSeek-V4-Flash	60.98	59.69	**82.94**	**82.89**	**62.44**	**78.12**
GLM-5.2	67.46	65.95	71.70	70.26	54.83	77.19
Kimi-K2.6	49.79	48.85	50.41	64.29	42.27	62.62
Qwen3.5-397B-A17B	33.35	14.36	29.07	35.92	22.97	36.35
Qwen3.6-35B-A3B	46.72	24.23	44.15	57.53	27.48	51.54
CompoWorld (ours)	73.28	**73.63**	76.66	79.02	58.93	74.54
Δ vs. backbone	+26.56	+49.40	+32.51	+21.49	+31.45	+23.00

F Environment Quality and Selective World-Model Simulation

In the cases we inspected, the synthesized environments exhibited good practical quality overall. Manual spot checks showed that the tool responses and state changes were generally consistent with the intended service behavior and supported coherent task execution. This was partly due to the additional usability checks we incorporated into the environment construction process: each deterministic implementation must pass automatically generated tests covering basic scenarios such as normal invocation and error handling. Such tests cannot cover the full complexity of real services. Real systems often contain more complex business rules, implicit dependencies, and less common boundary cases, which are difficult to fully capture through finite tests. Current coding agents cannot fully resolve this limitation. We expect that as coding agents become more capable, their ability to generate and validate tool implementations will correspondingly strengthen, thereby continuously improving the quality of synthesized environments. Since it is difficult to directly verify the usability of all synthesized environments through large-scale evaluation of individual environments, we primarily evaluate their overall utility through the performance gains brought by downstream training. Although this evaluation cannot directly prove that every environment instance is completely accurate, sustained downstream gains can indirectly indicate that these environments are generally able to provide effective and usable training signals.

In addition, for tools in our experiments that cannot be directly mocked through local implementations, we adopt world-model simulation selectively. Specifically, only 7.1% of tools rely on world-model simulation, and only about 10% of samples invoke at least one such tool. Therefore, the vast majority of tools and training samples still rely on deterministic execution, which limits the proportion of the training corpus directly exposed to simulation errors. These proportions do not imply that simulation errors can be ignored in the affected workflows: once inaccurate observations are generated, they may still affect the agent's subsequent judgments and decisions. The main purpose of our selective simulation is to strike a balance between environment fidelity and capability coverage. Some

operations depend on external services, model inference, or specialized runtimes, and are therefore difficult to faithfully reproduce through existing local implementations. Removing these tools would reduce the capabilities covered by the environment and eliminate complete workflows that depend on them. For these tools, we use a world model to approximately generate tool responses and necessary state updates based on the current service state and executed action. Prior work also provides some support for this approach. Qwen-AgentWorld (Zuo et al., 2026) models environment responses across seven interaction domains, including MCP tools, and reports downstream agent performance improvements brought by training with simulated environments. Its evaluation also compares simulated behavior with real interaction references, providing experimental evidence that language world models can provide effective training feedback. These results indicate that using world models to expand environment coverage is a practically valuable approach. In CompoWorld, world-model simulation provides a limited complement to tested deterministic tools.

G Executable Environment Example

Google Calendar illustrates how an executable environment supports scheduling tasks. Events are represented as typed records, and tools retrieve or modify their fields while returning structured observations. The example below shows the event state and the update operation used to move a community meetup online. The code retains the complete update logic; shared helpers and unrelated methods are omitted for readability.

Google Calendar mock: typed state and an executable update

```python
class Event(BaseRecord):
    summary: str
    start: EventTime
    end: EventTime
    description: Optional[str] = None
    location: Optional[str] = None

class GoogleCalendarState(BaseModel):
    events: list[Event] = Field(default_factory=list)
    # Shared record lookup methods are omitted here.

def update_event(world: GoogleCalendarState, **kwargs) -> str:
    if "eventId" not in kwargs:
        return json.dumps({"success": False, "error": {
            "code": 400, "message": "Missing required parameter: eventId
"}})

    event = world.get_by_id("events", kwargs["eventId"])
    if event is None:
        return json.dumps({"success": False, "error": {
            "code": 404, "message": f"Event not found: {kwargs['eventId
']}"}})

    updatable_fields = ("summary", "description", "location", "start", "
end")
    for field in updatable_fields:
        if field in kwargs:
            value = kwargs[field]
            if field in ("start", "end"):
                value = EventTime(**value)
            setattr(event, field, value)

    event.updated_at = _now()
    return json.dumps({"success": True, "data": event.model_dump(mode="
json")})
```

Observed state transition in the training sample

The trajectory in Appendix H invokes `update_event(eventId="<meetup-event>", location="Online")`.

Field	Before the call	Returned observation
Location	Khangarid Hall, Ulaanbaatar	**Online**
Date and time	Jan. 20, 18:30–21:00 (UTC+8)	Unchanged
Result	Existing event in local state	`success: true`

The same interface supports both successful transitions and explicit failure observations: a missing identifier returns error 400, while an unknown event returns error 404. Only supplied, supported fields are changed, allowing the task rubric to inspect the resulting event state.

H A Complete Agent Trajectory

This example follows an agent preparing a community anniversary meetup. Google Calendar holds the event; the mail and collaboration environment (Simosphere) provides the runbook, volunteer roster, inventory, and meeting invitations. Reddit hosts the announcement and volunteer replies, while Thinair Geo supplies the venue forecast. The task connects these environments through two conditions: swag-exempt volunteers should not receive a pin-reservation reply, and snow in the forecast requires moving the meetup online while preserving its time and inviting all confirmed volunteers.

The interaction below retains the user request, every tool call and its arguments, and the final response. Observations are abridged to emphasize tool capabilities, task-relevant evidence, and state changes; repeated operating guidance and administrative metadata are omitted. We also omit the agent's reasoning content. Descriptive aliases replace identifiers and contact details consistently throughout.

Task setting

Today's date is 2026-01-15.

You are an AI assistant that completes user tasks by calling tools.

You have access to a set of tools exposed via the OpenAI function-calling interface. To make progress, emit a tool_call. Observe the JSON result (each tool returns `{"success": true, "data": ...}` or '{"success": false, "error": {"code": ..., "message": ...}}') and decide the next step.

Rules:

- Never invent tool names or arguments. Only use the tools you were given.
- If a required argument is missing, do NOT make it up - call a read tool first to discover the value (e.g. list_issues to find an issue number before update_issue).
- Correct 400 errors by supplying the missing arg on the next turn. Correct 404 errors by checking your assumptions about ids/names.
- When the task is complete, respond with a short natural-language confirmation to the user WITHOUT any tool_calls - that ends the session.
- Keep going until you have finished the whole task. Do not stop early.

How tools work in this session

You do NOT have the real tools registered. You cannot see their names or argument schemas ahead of time, and you cannot call them by name directly. Instead four meta-tools are your only interface to the environment:

- list_services() - the services (environments) mounted in this session. You start knowing nothing but their names.
- list_tools(service) - every tool on a service, with name and description,

Task setting (continued)

but NOT its argument schema.
• describe_tool(tool) - the full argument schema of one tool. Call this before calling a tool whose arguments you do not know.
• call_tool(tool_name, arguments) - execute a real tool and return its result. ‘`arguments`‘ is a JSON object matching the schema from describe_tool. You are responsible for formatting it correctly; there is no validation beyond what the tool itself enforces, so a wrong shape will simply fail.

Workflow: list_services -> list_tools on a service that looks relevant -> describe_tool for the tool you want to use -> call_tool to run it. Re-check with list_tools/describe_tool whenever you need a tool you have not yet examined; the surface is larger than you can hold in one glance, and the right tool for a later step often lives on a service you have not looked at yet. Do not guess tool names or argument fields - look them up.

User request

Frostbyte’s 5th-anniversary community meetup is coming up on Tuesday, January 20, and I’m swamped - please run the whole prep for me. Start from the ’Frostbyte 5th Anniversary Community Meetup’ event on my Google Calendar; it points you to the prep runbook Dara sent to the Frostbyte shared mailbox. Follow that runbook end to end. In particular: post a comment in the pinned announcement thread on r/frostbyte telling the community how many anniversary pins are on hand; reply to every confirmed volunteer’s comment in that thread to tell them their pin is reserved, except volunteers the roster flags as swag-exempt (they already have theirs); and handle the weather contingency - if snow shows up anywhere in the venue’s forecast between now and the meetup, the meetup moves online: post a comment in the thread saying so, set the calendar event’s location to ’Online’, and create an online meeting invite for the confirmed volunteers for the same date and time. If there’s no snow in the forecast, keep it in person: post a reminder comment with the venue address and leave the calendar event as it is. Finally, reply to Dara’s runbook email with a short wrap-up, and tell me what you did.

Turn 1: Discover the available environments

Assistant
I’ll start by exploring the environment to understand what services and tools are available.

Action 1 | list_services

```
{"name": "list_services", "arguments": {}}
```

Observation 1 | list_services

```
{
  "services": ["reddit", "shopify", "google_calendar", "thinair_geo", "
  simosphere", "chirona", "buyapi", "tableau", "fractera", "servicem8", "
  promptibus", "webex", "discord", "polymarket", "cometchat"]
}
```

Turn 2: Inspect calendar, community, and weather tools

Assistant
Let me look at the tools available on the most relevant services first.

Action 2 | list_tools

```
{"name": "list_tools", "arguments": {"service": "google_calendar"}}
```

Action 3 | list_tools

Turn 2: Inspect calendar, community, and weather tools (continued)

```
{"name": "list_tools", "arguments": {"service": "reddit"}}
```

Action 4 | list_tools

```
{"name": "list_tools", "arguments": {"service": "thinair_geo"}}
```

Observation 2 | list_tools


```
{
  "service": "google_calendar",
  "tools": [
    {"name": "google_calendar__create_event", "description": "Creates a new
   event in Google Calendar."},
    {"name": "google_calendar__get_event", "description": "Retrieves
  details of a specific event by its ID."},
    {"name": "update_event", "description": "Updates an existing event in
  Google Calendar."},
    {"name": "google_calendar__delete_event", "description": "Deletes an
  event from Google Calendar."},
    {"name": "google_calendar__list_events", "description": "Lists events
  within a specified time range from Google Calendar."}
  ]}
```


Observation 3 | list_tools


```
{
  "service": "reddit",
  "tools": [
    {"name": "reddit_create_reddit_post", "description": "Creates a new
  text or link post on a specified, existing Reddit subreddit, optionally
  applying a flair."},
    {"name": "reddit_delete_reddit_comment", "description": "Deletes a
  Reddit comment, identified by its fullname ID, if it was authored by the
  authenticated user."},
    {"name": "reddit_delete_reddit_post", "description": "Permanently
  deletes a Reddit post by its ID, provided the authenticated user has
  deletion permissions for that post."},
    {"name": "reddit_edit_reddit_comment_or_post", "description": "Edits
  the body text of the authenticated user's own existing comment or self-
  post on Reddit; cannot edit link posts or titles."},
    {"name": "reddit_get_user_flair", "description": "Fetches the list of
  available link flairs (i.e., post flairs, not user flairs) for a given
  subreddit."},
    {"name": "reddit_post_reddit_comment", "description": "Posts a comment
  on Reddit, replying to an existing and accessible submission (post) or
  another comment."},
    {"name": "reddit_retrieve_post_comments", "description": "Retrieves all
   comments for a Reddit post given its article ID (which must be for an
  existing, public post); nested replies within comments are returned as
  raw dictionaries requiring parsing."},
    {"name": "reddit_retrieve_reddit_post", "description": "Retrieves the
  current hot posts from a specified, publicly accessible subreddit."},
    {"name": "reddit_retrieve_specific_comment", "description": "Retrieves
  detailed information for a specific Reddit comment or post using its
  fullname."},
    {"name": "reddit_search_across_subreddits", "description": "Searches
  Reddit for content (e.g., posts, comments) using a query, with results
  typically confined to subreddits unless `restrict_sr` is set to False."}
  ]}
```


Observation 4 | list_tools


```
{
```

Turn 2: Inspect calendar, community, and weather tools (continued)


```
  "service": "thinair_geo",
  "tools": [
    {"name": "geocode", "description": "Convert an address, place name,
  street, or intersection into coordinates and structured location results
  ."},
    {"name": "reverse_geocode", "description": "Convert coordinates into
  the nearest address, street, or place."},
    {"name": "directions", "description": "Generate routes, ETAs, and turn-
  by-turn directions between locations."},
    {"name": "traffic", "description": "Retrieve live traffic conditions,
  congestion, and speed for a location."},
    {"name": "weather", "description": "Get current and forecast weather
  for a location, including severe weather alerts and minute-by-minute
  precipitation."},
    {"name": "isochrone", "description": "Generate travel-time or travel-
  distance reachability polygons from an origin."},
    {"name": "search_places", "description": "CATEGORY-specific POI search
  near a point  -  gas stations, truck stops, restaurants, charging
  stations, etc."},
    {"name": "batch_geocode", "description": "Geocode multiple addresses in
   one request with structured per-record results."},
    {"name": "explore", "description": "BROWSING / DISCOVERY search  -
  cities, neighbourhoods, or mixed venues near a location."},
    {"name": "quota", "description": "Check current usage, remaining limits
  , plan, and quota breakdown for the caller."},
    {"name": "issue_api_key", "description": "Mint a fresh API key for your
   current authenticated user/tenant."}
  ]}
```


Turn 3: Search for mail and roster tools

Assistant

Let me examine the remaining services to find email, roster, and other relevant tools.

Action 5 | list_tools


```
{"name": "list_tools", "arguments": {"service": "chirona"}}
```


Action 6 | list_tools


```
{"name": "list_tools", "arguments": {"service": "promptibus"}}
```


Action 7 | list_tools


```
{"name": "list_tools", "arguments": {"service": "webex"}}
```


Action 8 | list_tools


```
{"name": "list_tools", "arguments": {"service": "cometchat"}}
```


Observation 5 | list_tools


```
{
  "service": "chirona",
  "tool_names": ["conversation_initialisation_critical_instructions", "
  get_athlete_stats", "get_strava_recent_activities", "
  get_strava_latest_run", "get_strava_activity_details", "
  get_recent_activities", "get_latest_run", "get_activity_details", "
  get_period_summary", "get_latest_sleep", "get_latest_body", "
  get_latest_daily", "push_planned_workout", "get_planned_workouts", "
  delete_planned_workout"]
}
```


Observation 6 | list_tools

Turn 3: Search for mail and roster tools (continued)


```
{
  "service": "promptibus",
  "tool_names": ["recommend_model", "optimize_prompt", "lint_prompt", "
  compare_models", "get_parameters", "get_model_profile", "get_pricing"]
}
```


Observation 7 | list_tools


```
{
  "service": "webex",
  "tool_names": ["webex_create_room", "webex_create_team", "
  webex_get_team_details", "webex_list_teams", "webex_list_webhooks", "
  webex_messaging_create_message", "webex_messaging_create_room", "
  webex_messaging_create_team_membership", "
  webex_messaging_delete_membership", "webex_messaging_delete_message", "
  webex_messaging_delete_room", "webex_messaging_get_membership_details", "
  webex_messaging_get_message_details", "webex_messaging_get_room_details",
   "webex_messaging_get_team_membership_details", "
  webex_messaging_list_memberships", "webex_messaging_list_messages", "
  webex_messaging_list_rooms", "webex_messaging_list_team_memberships", "
  webex_people_get_person", "webex_people_list_people", "
  webex_rooms_get_room_details", "webex_update_membership", "
  webex_update_room", "webex_update_team", "webex_webhooks_create_webhook",
   "webex_webhooks_delete_webhook", "webex_webhooks_get_webhook"]
}
```


Observation 8 | list_tools


```
{
  "service": "cometchat",
  "tool_names": ["search_cometchat_docs", "fetch_cometchat_doc_page", "
  get_cometchat_implementation_bundle"]
}
```


Turn 4: Complete the service inspection

Assistant

Let me check the remaining services for email and roster capabilities.

Action 9 | list_tools


```
{"name": "list_tools", "arguments": {"service": "shopify"}}
```


Action 10 | list_tools


```
{"name": "list_tools", "arguments": {"service": "simosphere"}}
```


Action 11 | list_tools


```
{"name": "list_tools", "arguments": {"service": "buyapi"}}
```


Action 12 | list_tools


```
{"name": "list_tools", "arguments": {"service": "tableau"}}
```


Action 13 | list_tools


```
{"name": "list_tools", "arguments": {"service": "fractera"}}
```


Action 14 | list_tools


```
{"name": "list_tools", "arguments": {"service": "servicem8"}}
```


Action 15 | list_tools


```
{"name": "list_tools", "arguments": {"service": "discord"}}
```

Turn 4: Complete the service inspection (continued)

Action 16 | list_tools


```
{"name": "list_tools", "arguments": {"service": "polymarket"}}
```


Observation 9 | list_tools


```
{"service": "shopify", "tools": []}
```


Observation 10 | list_tools


```
{
  "service": "simosphere",
  "tools": [
    {"name": "identity_me", "description": "Retrieve the profile of the
  connected Microsoft 365 user, including id, display name, email address,
  job title, office location, and user principal name."},
    {"name": "mail_list", "description": "List the most recent messages in
  a mail folder of the connected mailbox."},
    {"name": "mail_search", "description": "Search the connected mailbox
  using KQL (Keyword Query Language) syntax."},
    {"name": "mail_read", "description": "Read a single email message by
  its ID, including the full HTML body, sender, recipients (to and cc),
  received date, and attachment indicator."},
    {"name": "mail_reply", "description": "Reply to an existing email
  message on behalf of the connected mailbox owner."},
    {"name": "calendar_list", "description": "List upcoming calendar events
   for the connected mailbox owner within a time window."},
    {"name": "calendar_create", "description": "Create a new calendar event
   on the connected mailbox owner's primary calendar."},
    {"name": "files_list", "description": "List files and folders in the
  connected mailbox owner's OneDrive."},
    {"name": "files_search", "description": "Search the connected mailbox
  owner's OneDrive for files and folders by name or content."},
    {"name": "people_search", "description": "Search contacts and the
  Microsoft Entra ID directory from the connected mailbox owner's
  perspective."},
    {"name": "people_mymanager", "description": "Retrieve the direct
  manager of the connected mailbox owner as configured in Microsoft Entra
  ID (Azure AD)."},
    {"name": "site_metadata", "description": "Retrieve metadata for the
  connected SharePoint site, including display name, web URL, description,
  creation and last-modified dates, and site collection details."},
    {"name": "site_lists", "description": "List all SharePoint lists and
  document libraries on the connected site."},
    {"name": "site_list_items", "description": "List items from a specific
  SharePoint list on the connected site, with all custom fields expanded
  ."},
    {"name": "site_search", "description": "Search the connected SharePoint
   site for matching pages, list items, and files using the Microsoft
  Search API."},
    {"name": "site_drive_list", "description": "List files and folders in
  the connected SharePoint site's default document library."},
    {"name": "site_drive_read", "description": "Get detailed metadata and
  thumbnail URLs for a single file or folder in the connected SharePoint
  site's default document library."},
    {"name": "site_drive_upload", "description": "Upload a small file (
  maximum 4 MB) to a folder in the connected SharePoint site's default
  document library."},
    {"name": "site_drive_share", "description": "Create a sharing link for
  a file or folder in the connected SharePoint site's default document
  library."},
```

Turn 4: Complete the service inspection (continued)


```
    {"name": "site_pages_list", "description": "List all modern SharePoint
  pages published on the connected site."}
  ]}
```


Observation 11 | list_tools


```
{
  "service": "buyapi",
  "tool_names": ["vendors_resolve", "vendors_details", "stacks_findsimilar
  ", "vendors_compare", "vendors_estimatecost", "stacks_recommend"]
}
```


Observation 12 | list_tools


```
{
  "service": "tableau",
  "tool_names": ["list_datasources", "get_datasource_metadata", "
  query_datasource", "query_datasource_fixed", "list_fields", "
  list_workbooks", "get_workbook", "list_views", "get_view_data", "
  get_view_image", "get_custom_view_image", "search_content", "
  list_projects", "list_users", "list_extract_refresh_tasks", "
  get_pulse_metric", "get_pulse_metric_definition", "get_pulse_subscription
  ", "get_pulse_metric_value_insight_bundle", "get_stale_content_report"]
}
```


Observation 13 | list_tools


```
{
  "service": "fractera",
  "tool_names": ["register_and_deploy", "check_status", "get_subdomain", "
  retry_deploy", "get_vps_recommendation", "get_project_info"]
}
```


Observation 14 | list_tools


```
{
  "service": "servicem8",
  "tool_names": ["servicem8_create_job_payment", "servicem8_delete_job_note
  ", "servicem8_delete_job_payment", "servicem8_list_all_assets", "
  servicem8_list_all_clients", "servicem8_list_all_document_templates", "
  servicem8_list_all_forms", "servicem8_list_all_job_notes", "
  servicem8_list_all_job_queues", "servicem8_list_all_jobs", "
  servicem8_list_all_locations", "servicem8_list_all_materials", "
  servicem8_list_all_tasks", "servicem8_retrieve_client", "
  servicem8_retrieve_form", "servicem8_retrieve_job", "
  servicem8_retrieve_job_activity", "servicem8_retrieve_job_note", "
  servicem8_retrieve_job_payment", "servicem8_retrieve_job_queue", "
  servicem8_retrieve_location", "servicem8_retrieve_material", "
  servicem8_retrieve_staff_member", "servicem8_update_job_note", "
  servicem8_update_job_payment"]
}
```


Observation 15 | list_tools


```
{
  "service": "discord",
  "tool_names": ["get_server_info", "list_servers", "get_server_members", "
  get_user_info", "set_nickname", "get_member_roles", "send_message", "
  edit_message", "delete_message", "get_messages", "pin_message", "
  unpin_message", "get_pinned_messages", "add_reaction", "list_channels", "
  get_channel_info", "create_text_channel", "create_voice_channel", "
  edit_channel", "delete_channel", "set_channel_topic", "
  set_channel_slowmode", "set_channel_nsfw", "get_channel_invites", "
  list_categories", "create_category", "edit_category", "delete_category",
```

Turn 4: Complete the service inspection (continued)


```
  "create_webhook", "list_webhooks", "edit_webhook", "delete_webhook", "
  send_webhook_message", "list_roles", "create_role", "edit_role", "
  delete_role", "add_role", "remove_role", "ban_member", "unban_member", "
  kick_member", "timeout_member", "remove_timeout", "get_bans", "
  get_voice_states", "move_member_to_voice", "disconnect_member_from_voice
  ", "discord__list_events", "discord__create_event", "edit_event", "
  discord__delete_event", "get_event_users", "get_channel_permissions", "
  set_channel_permissions", "get_role_permissions", "set_role_permissions",
   "create_invite", "list_server_invites", "delete_invite", "
  create_forum_channel", "create_forum_post", "list_forum_posts", "
  edit_forum_post", "lock_forum_post", "list_emojis", "create_emoji", "
  edit_emoji", "delete_emoji", "create_thread", "list_active_threads", "
  join_thread", "archive_thread"]
}
```


Observation 16 | list_tools


```
{
  "service": "polymarket",
  "tool_names": ["search_markets", "get_market", "search_events", "
  polymarket__get_event", "list_tags", "get_trades", "analyze_market"]
}
```


Turn 5: Inspect the calendar and email interfaces

Assistant

The email service is `simosphere` (Microsoft 365). Let me look at the key tools I'll need: calendar listing, mail search/read, and check the schema details.

Action 17 | google_calendar__list_events


```
{"name": "describe_tool", "arguments": {"tool": "
  google_calendar__list_events"}}
```


Action 18 | mail_search


```
{"name": "describe_tool", "arguments": {"tool": "mail_search"}}
```


Action 19 | mail_read


```
{"name": "describe_tool", "arguments": {"tool": "mail_read"}}
```


Action 20 | mail_reply


```
{"name": "describe_tool", "arguments": {"tool": "mail_reply"}}
```


Observation 17 | google_calendar__list_events


```
{
  "name": "google_calendar__list_events",
  "description": "Lists events within a specified time range from Google
  Calendar.",
  "parameters": {
    "type": "object",
    "properties": {
      "timeMin": {"type": "string"},
      "timeMax": {"type": "string"},
      "maxResults": {"type": "integer"},
      "orderBy": {"type": "string", "enum": ["startTime", "updated"]}
    },
    "required": ["timeMin", "timeMax"]
  }
}
```

Turn 5: Inspect the calendar and email interfaces (continued)

Observation 18 | mail_search


```
{
  "name": "mail_search",
  "description": "Search the connected mailbox using KQL (Keyword Query
  Language) syntax.",
  "parameters": {
    "type": "object",
    "properties": {"top": {"type": "number", "default": 20}, "query": {"
  type": "string"}},
    "required": ["query"]
  }
}
```


Observation 19 | mail_read


```
{
  "name": "mail_read",
  "description": "Read a single email message by its ID, including the full
   HTML body, sender, recipients (to and cc), received date, and attachment
   indicator.",
  "parameters": {"type": "object", "properties": {"messageId": {"type": "
  string"}}, "required": ["messageId"]}
}
```


Observation 20 | mail_reply


```
{
  "name": "mail_reply",
  "description": "Reply to an existing email message on behalf of the
  connected mailbox owner.",
  "parameters": {
    "type": "object",
    "properties": {
      "bodyHtml": {"type": "string"},
      "replyAll": {"type": "boolean", "default": false},
      "messageId": {"type": "string"}
    },
    "required": ["bodyHtml", "messageId"]
  }
}
```


Turn 6: Retrieve the event and runbook

Assistant

Now let me find the calendar event and search the mailbox for Dara's runbook email in parallel.

Action 21 | google_calendar__list_events


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "google_calendar__list_events",
    "arguments": {
      "timeMin": "2026-01-15T00:00:00Z",
      "timeMax": "2026-01-25T23:59:59Z",
      "orderBy": "startTime"
    }
  }
}
```


Action 22 | mail_search

Turn 6: Retrieve the event and runbook (continued)

```
{
  "name": "call_tool",
  "arguments": {"tool_name": "mail_search", "arguments": {"query": "runbook
  "}}
}
```

Observation 21 | google_calendar__list_events


```
{
  "success": true,
  "data": [
    {
      "id": "<standup-event>",
      "summary": "Frostbyte core team standup",
      "start": {"dateTime": "2026-01-16T09:30:00+08:00", "timeZone": "Asia/
  Ulaanbaatar"},
      "end": {"dateTime": "2026-01-16T10:00:00+08:00", "timeZone": "Asia/
  Ulaanbaatar"},
      "description": "Weekly sync.",
      "location": null
    },
    {
      "id": "<release-event>",
      "summary": "Frostbyte v3.2 release freeze",
      "start": {"dateTime": "2026-01-19T09:00:00+08:00", "timeZone": "Asia/
  Ulaanbaatar"},
      "end": {"dateTime": "2026-01-19T10:00:00+08:00", "timeZone": "Asia/
  Ulaanbaatar"},
      "description": "No new commits on main until the anniversary release
  is out.",
      "location": null
    },
    {
      "id": "<meetup-event>",
      "summary": "Frostbyte 5th Anniversary Community Meetup",
      "start": {"dateTime": "2026-01-20T18:30:00+08:00", "timeZone": "Asia/
  Ulaanbaatar"},
      "end": {"dateTime": "2026-01-20T21:00:00+08:00", "timeZone": "Asia/
  Ulaanbaatar"},
      "description": "Five years of Frostbyte! Catering, lightning talks,
  anniversary swag. In-person at Khangarid Hall, 42 Peace Avenue,
  Ulaanbaatar. Prep for this event is run by the runbook: search the
  Frostbyte shared mailbox for Dara's 'Meetup runbook' email and follow it
  end to end.",
      "location": "Khangarid Hall, 42 Peace Avenue, Ulaanbaatar"
    }
  ]
}
```


Observation 22 | mail_search


```
{
  "success": true,
  "data": [
    {
      "id": "<runbook-email>",
      "subject": "Meetup runbook - 5th anniversary prep (please follow it
  end to end)",
      "sender": "<dara-email>",
      "to_recipients": ["<nadia-email>"],
      "cc_recipients": [],
```

Turn 6: Retrieve the event and runbook (continued)


```
      "body_html": "<p>Hi Nadia,</p><p>Since you’re running anniversary
    prep this week, here’s the runbook we agreed at the core-team sync &mdash
    ; please work through it end to end and reply to this email with a short
    wrap-up when you’re done.</p><ol><li><b>Anniversary pins.</b> The
    anniversary pins arrived from the mint. Post a comment on the pinned
    announcement thread in r/frostbyte (&quot;Announcement: Frostbyte 5th
    Anniversary Community Meetup &mdash; Tue Jan 20&quot;) telling the
    community how many anniversary pins are on hand &mdash; the count is in
    the &quot;Meetup Details&quot; list on the Frostbyte Community Hub (our
    SharePoint site).</li><li><b>Volunteer pins.</b> Volunteer sign-ups are
    in the comments of that same announcement thread, and the post itself
    says where each volunteer’s status is tracked. Reply to every comment in
    the thread from a volunteer whose status is <b>Confirmed</b>, telling
    them their anniversary pin is reserved &mdash; but skip anyone the roster
     flags as swag_exempt = Yes (they already collected their pin at FOSDEM)
    .</li><li><b>Weather contingency.</b> The venue’s note in the
    announcement thread comments has the venue’s map-pin coordinates. Pull
    the weather forecast for those coordinates covering the days up to and
    including the meetup (Jan 20). <b>If snow shows up anywhere in that
    forecast, the meetup moves online:</b> post a comment on the announcement
     thread saying it’s going online, set the calendar event’s location to &
    quot;Online&quot; (the &quot;Frostbyte 5th Anniversary Community Meetup&
    quot; event on the project Google Calendar), and create an online meeting
     invite on the shared-workspace calendar for the same date and time as
    the meetup with the confirmed volunteers as attendees. <b>If there is no
    snow in that forecast</b>, keep it in person: post a reminder comment
    with the venue address and leave the calendar event as it is.</li><li><b>
    Wrap-up.</b> When all of the above is done, reply to this email with a
    short summary of what you did.</li></ol><p>Thanks!<br>&mdash; Dara</p>",
      "status": "received"
    }
  ]
}
```


Turn 7: Inspect community and roster interfaces

Assistant

Found the calendar event and the runbook email. The runbook references:
1. A pinned announcement thread on r/frostbyte
2. A "Meetup Details" list on the Frostbyte Community Hub (SharePoint)
3. Venue coordinates in the thread comments for weather check

Let me examine the relevant tools for Reddit and SharePoint.

Action 23 | reddit_retrieve_reddit_post


```
{"name": "describe_tool", "arguments": {"tool": "
  reddit_retrieve_reddit_post"}}
```


Action 24 | reddit_retrieve_post_comments


```
{"name": "describe_tool", "arguments": {"tool": "
  reddit_retrieve_post_comments"}}
```


Action 25 | reddit_post_reddit_comment


```
{"name": "describe_tool", "arguments": {"tool": "reddit_post_reddit_comment
  "}}
```


Action 26 | site_lists


```
{"name": "describe_tool", "arguments": {"tool": "site_lists"}}
```

Turn 7: Inspect community and roster interfaces (continued)

Action 27 | site_list_items


```
{"name": "describe_tool", "arguments": {"tool": "site_list_items"}}
```


Action 28 | identity_me


```
{"name": "describe_tool", "arguments": {"tool": "identity_me"}}
```


Observation 23 | reddit_retrieve_reddit_post


```
{
  "name": "reddit_retrieve_reddit_post",
  "description": "Retrieves the current hot posts from a specified,
  publicly accessible subreddit.",
  "parameters": {
    "type": "object",
    "properties": {
      "size": {"type": "integer", "default": 5, "nullable": true},
      "subreddit": {"type": "string"}
    },
    "required": ["subreddit"]
  }
}
```


Observation 24 | reddit_retrieve_post_comments


```
{
  "name": "reddit_retrieve_post_comments",
  "description": "Retrieves all comments for a Reddit post given its
  article ID (which must be for an existing, public post); nested replies
  within comments are returned as raw dictionaries requiring parsing.",
  "parameters": {"type": "object", "properties": {"article": {"type": "
  string"}}, "required": ["article"]}
}
```


Observation 25 | reddit_post_reddit_comment


```
{
  "name": "reddit_post_reddit_comment",
  "description": "Posts a comment on Reddit, replying to an existing and
  accessible submission (post) or another comment.",
  "parameters": {
    "type": "object",
    "properties": {"text": {"type": "string"}, "thing_id": {"type": "string
  "}},
    "required": ["text", "thing_id"]
  }
}
```


Observation 26 | site_lists


```
{
  "name": "site_lists",
  "description": "List all SharePoint lists and document libraries on the
  connected site.",
  "parameters": {"type": "object", "properties": {"top": {"type": "number",
   "default": 50}}}
}
```


Observation 27 | site_list_items


```
{
  "name": "site_list_items",
  "description": "List items from a specific SharePoint list on the
  connected site, with all custom fields expanded.",
```

Turn 7: Inspect community and roster interfaces (continued)


```
  "parameters": {
    "type": "object",
    "properties": {"top": {"type": "number", "default": 50}, "listId": {"
  type": "string"}},
    "required": ["listId"]
  }
}
```


Observation 28 | identity_me


```
{
  "name": "identity_me",
  "description": "Retrieve the profile of the connected Microsoft 365 user,
   including id, display name, email address, job title, office location,
  and user principal name.",
  "parameters": {"type": "object", "properties": {}}
}
```


Turn 8: Locate the announcement, lists, and mailbox

Assistant

Now let me find the announcement thread on Reddit, the SharePoint list, and confirm the mailbox identity - all in parallel.

Action 29 | reddit_retrieve_reddit_post


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "reddit_retrieve_reddit_post",
    "arguments": {"subreddit": "frostbyte", "size": 25}
  }
}
```


Action 30 | site_lists


```
{"name": "call_tool", "arguments": {"tool_name": "site_lists", "arguments":
   {}}}
```


Action 31 | identity_me


```
{"name": "call_tool", "arguments": {"tool_name": "identity_me", "arguments
  ": {}}}
```


Observation 29 | reddit_retrieve_reddit_post


```
{
  "success": true,
  "data": [
    {
      "id": "<announcement>",
      "kind": "self",
      "title": "Announcement: Frostbyte 5th Anniversary Community Meetup -
  Tue Jan 20",
      "text": "Hi r/frostbyte! Frostbyte turns five, and we're celebrating
  with a community meetup on Tuesday, January 20, 18:30-21:00 (UTC+8) at
  Khangarid Hall, 42 Peace Avenue, Ulaanbaatar. Catering is on us and there
   will be lightning-talk slots. If you signed up as a volunteer, your
  status (Confirmed / Pending / Declined) is tracked in the Volunteer
  Roster list on the Frostbyte Community Hub - our SharePoint site.
  Confirmed volunteers get the 5th anniversary enamel pin. See the comments
   below for the venue note from the hall. - Dara",
      "url": null,
```

Turn 8: Locate the announcement, lists, and mailbox (continued)


```
      "subreddit": "frostbyte",
      "score": 42
    },
    {
      "id": "<release-post>",
      "kind": "self",
      "title": "Frostbyte CLI 3.2.0 released",
      "text": "Faster config parsing, new `frostbyte dump` command, and
  better Windows support. Changelog inside.",
      "url": null,
      "subreddit": "frostbyte",
      "score": 88
    },
    {
      "id": "<help-post>",
      "kind": "self",
      "title": "How do I configure the new collector?",
      "text": "The docs mention a collector config but I can't find the
  reference. Anyone got an example?",
      "url": null,
      "subreddit": "frostbyte",
      "score": 3
    }
  ]
}
```


Observation 30 | site_lists


```
{
  "success": true,
  "data": [
    {
      "id": "<volunteer-roster>",
      "name": "Volunteer Roster",
      "display_name": "Volunteer Roster",
      "description": "Volunteers for the 5th anniversary meetup, with RSVP
  status and swag flags.",
      "list_type": "generic"
    },
    {
      "id": "<meetup-details>",
      "name": "Meetup Details",
      "display_name": "Meetup Details",
      "description": "Logistics for the anniversary meetup: swag shipping,
  budget, catering.",
      "list_type": "generic"
    },
    {
      "id": "<talk-slots>",
      "name": "Lightning Talk Slots",
      "display_name": "Lightning Talk Slots",
      "description": "Proposed lightning talk slots for the meetup.",
      "list_type": "generic"
    }
  ]
}
```


Observation 31 | identity_me


```
{
  "success": true,
```

Turn 8: Locate the announcement, lists, and mailbox (continued)


```
  "data": {
    "id": "<organizer-profile>",
    "display_name": "Nadia Okafor",
    "email": "<nadia-email>",
    "job_title": "Community Lead",
    "office_location": "Ulaanbaatar",
    "user_principal_name": "<nadia-email>",
    "company_name": "Frostbyte OSS"
  }
}
```


Turn 9: Read comments, volunteers, and meetup details

Assistant

Found everything. Now let me get the announcement thread comments, the Volunteer Roster, and the Meetup Details list - in parallel.

Action 32 | reddit_retrieve_post_comments

```
{
  "name": "call_tool",
  "arguments": {"tool_name": "reddit_retrieve_post_comments", "arguments":
  {"article": "<announcement>"}}
}
```

Action 33 | site_list_items

```
{
  "name": "call_tool",
  "arguments": {"tool_name": "site_list_items", "arguments": {"listId": "<
  volunteer-roster>"}}
}
```

Action 34 | site_list_items

```
{
  "name": "call_tool",
  "arguments": {"tool_name": "site_list_items", "arguments": {"listId": "<
  meetup-details>"}}
}
```

Observation 32 | reddit_retrieve_post_comments


```
{
  "success": true,
  "data": [
    {
      "id": "<maya-comment>",
      "text": "Maya Chen here (/u/<maya-account>) - Confirmed! Can't wait,
  I'll bring the collector demo.",
      "parent_id": "t3_<announcement>"
    },
    {
      "id": "<jonas-comment>",
      "text": "Jonas Weber (/u/<jonas-account>) - Confirmed, count me in. I
  'll take a lightning talk slot on the new CLI.",
      "parent_id": "t3_<announcement>"
    },
    {
      "id": "<tomas-comment>",
      "text": "Tomas Rivera (/u/<tomas-account>) - Confirmed! See you all
  there.",
```

Turn 9: Read comments, volunteers, and meetup details (continued)


```
      "parent_id": "t3_<announcement>"
    },
    {
      "id": "<sofia-comment>",
      "text": "Sofia Marchetti (/u/<sofia-account>) - sorry, I have to
  decline this time. Have a great one, everyone!",
      "parent_id": "t3_<announcement>"
    },
    {
      "id": "<alex-comment>",
      "text": "Alex Kim (/u/<alex-account>) - hoping to make it if work
  allows. Fingers crossed!",
      "parent_id": "t3_<announcement>"
    },
    {
      "id": "<venue-comment>",
      "text": "Khangarid Hall here (/u/<venue-account>) - venue note: we're
   at 42 Peace Avenue, Ulaanbaatar. Map pin: 47.9186, 106.9176. Free
  parking in the rear lot. Heads-up: if snow is in the forecast, the hill
  road up to the hall closes, so keep an eye on the weather before you set
  out.",
      "parent_id": "t3_<announcement>"
    }
  ]
}
```


Observation 33 | site_list_items


```
{
  "success": true,
  "data": [
    {
      "id": "<maya-roster-entry>",
      "list_id": "<volunteer-roster>",
      "fields": {
        "volunteer": "Maya Chen",
        "reddit_username": "<maya-account>",
        "email": "<maya-email>",
        "rsvp_status": "Confirmed",
        "swag_exempt": "No"
      }
    },
    {
      "id": "<jonas-roster-entry>",
      "list_id": "<volunteer-roster>",
      "fields": {
        "volunteer": "Jonas Weber",
        "reddit_username": "<jonas-account>",
        "email": "<jonas-email>",
        "rsvp_status": "Confirmed",
        "swag_exempt": "No"
      }
    },
    {
      "id": "<tomas-roster-entry>",
      "list_id": "<volunteer-roster>",
      "fields": {
        "volunteer": "Tomas Rivera",
        "reddit_username": "<tomas-account>",
        "email": "<tomas-email>",
```

Turn 9: Read comments, volunteers, and meetup details (continued)


```
        "rsvp_status": "Confirmed",
        "swag_exempt": "Yes"
      }
    },
    {
      "id": "<sofia-roster-entry>",
      "list_id": "<volunteer-roster>",
      "fields": {
        "volunteer": "Sofia Marchetti",
        "reddit_username": "<sofia-account>",
        "email": "<sofia-email>",
        "rsvp_status": "Declined",
        "swag_exempt": "No"
      }
    },
    {
      "id": "<priya-roster-entry>",
      "list_id": "<volunteer-roster>",
      "fields": {
        "volunteer": "Priya Raman",
        "reddit_username": "<priya-account>",
        "email": "<priya-email>",
        "rsvp_status": "Pending",
        "swag_exempt": "No"
      }
    }
  ]
}
```


Observation 34 | site_list_items


```
{
  "success": true,
  "data": [
    {
      "id": "<pin-count-entry>",
      "list_id": "<meetup-details>",
      "fields": {"item": "anniversary_pins_on_hand", "value": "12"}
    },
    {
      "id": "<shipping-entry>",
      "list_id": "<meetup-details>",
      "fields": {
        "item": "swag_ship_to",
        "value": "Frostbyte Labs, 12 Narniin Zam 1, Ulaanbaatar 14200,
  Mongolia"
      }
    },
    {
      "id": "<budget-entry>",
      "list_id": "<meetup-details>",
      "fields": {"item": "swag_budget_code", "value": "<swag-budget>"}
    }
  ]
}
```

Turn 10: Inspect weather and calendar-update interfaces

Assistant

I now have all the prep data:

- **Pins on hand:** 12
- **Confirmed volunteers:** Maya Chen (<maya-account>), Jonas Weber (<jonas-account>), Tomas Rivera (<tomas-account> - but swag_exempt=Yes, so skip pin reply)
- **Venue coordinates:** 47.9186, 106.9176

Next, I need to check the weather forecast. Let me look at the weather tool and the calendar tools I may need.

Action 35 | weather

```
{"name": "describe_tool", "arguments": {"tool": "weather"}}
```

Action 36 | calendar_create

```
{"name": "describe_tool", "arguments": {"tool": "calendar_create"}}
```

Action 37 | update_event

```
{"name": "describe_tool", "arguments": {"tool": "update_event"}}
```

Action 38 | google_calendar__create_event

```
{"name": "describe_tool", "arguments": {"tool": "
  google_calendar__create_event"}}
```

Observation 35 | weather


```
{
  "name": "weather",
  "description": "Get current and forecast weather for a location,
  including severe weather alerts and minute-by-minute precipitation.",
  "parameters": {
    "type": "object",
    "properties": {
      "lat": {"type": "number", "description": "Latitude"},
      "lon": {"type": "number", "description": "Longitude"},
      "units": {
        "enum": ["imperial", "metric"],
        "type": "string",
        "description": "Temperature/wind units (default: imperial)"
      },
      "location": {"type": "string", "description": "Place name or address
  (will be geocoded)"},
      "forecast_days": {"type": "number", "description": "Forecast days
  1-16 (default 5)"},
      "include_alerts": {
        "type": "boolean",
        "description": "Include severe weather alerts and warnings (default
  : true)"
      },
      "include_forecast": {"type": "boolean", "description": "Include multi
  -day hourly forecast"},
      "include_minutely": {
        "type": "boolean",
        "description": "Include minute-by-minute precipitation for next 60
  min"
      }
    },
    "required": ["lat", "lon", "units", "location", "forecast_days", "
  include_alerts", "include_forecast", "include_minutely"]
  }
}
```

Turn 10: Inspect weather and calendar-update interfaces (continued)

Observation 36 | calendar_create


```
{
  "name": "calendar_create",
  "description": "Create a new calendar event on the connected mailbox
  owner's primary calendar.",
  "parameters": {
    "type": "object",
    "properties": {
      "endIso": {
        "type": "string",
        "format": "date-time",
        "description": "Event end time as ISO 8601 datetime (e.g.
  2024-12-01T10:00:00)"
      },
      "subject": {"type": "string", "description": "Title of the calendar
  event"},
      "bodyHtml": {
        "type": "string",
        "description": "Event description as HTML content, shown in the
  calendar invitation body"
      },
      "location": {
        "type": "string",
        "description": "Display name of the meeting location (e.g. \"
  Conference Room A\", \"Berlin Office\")"
      },
      "startIso": {
        "type": "string",
        "format": "date-time",
        "description": "Event start time as ISO 8601 datetime (e.g.
  2024-12-01T09:00:00)"
      },
      "timeZone": {
        "type": "string",
        "default": "Europe/Berlin",
        "description": "IANA time zone for start and end times (e.g. Europe
  /Berlin, America/New_York, UTC)"
      },
      "attendees": {
        "type": "array",
        "items": {"type": "string", "format": "email"},
        "default": [],
        "description": "List of attendee email addresses to invite"
      },
      "isOnlineMeeting": {
        "type": "boolean",
        "default": false,
        "description": "If true, provision a Microsoft Teams online meeting
   link for this event"
      }
    },
    "required": ["endIso", "subject", "bodyHtml", "location", "startIso"]
  }
}
```


Observation 37 | update_event


```
{
  "name": "update_event",
  "description": "Updates an existing event in Google Calendar.",
```

Turn 10: Inspect weather and calendar-update interfaces (continued)


```
  "parameters": {
    "type": "object",
    "properties": {
      "eventId": {"type": "string"},
      "summary": {"type": "string"},
      "start": {
        "type": "object",
        "properties": {"dateTime": {"type": "string"}, "timeZone": {"type":
 "string"}}
      },
      "end": {
        "type": "object",
        "properties": {"dateTime": {"type": "string"}, "timeZone": {"type":
 "string"}}
      },
      "description": {"type": "string"},
      "location": {"type": "string"}
    },
    "required": ["eventId"]
  }
}
```


Observation 38 | google_calendar__create_event


```
{
  "name": "google_calendar__create_event",
  "description": "Creates a new event in Google Calendar.",
  "parameters": {
    "type": "object",
    "properties": {
      "summary": {"type": "string"},
      "start": {
        "type": "object",
        "properties": {"dateTime": {"type": "string"}, "timeZone": {"type":
 "string"}},
        "required": ["dateTime"]
      },
      "end": {
        "type": "object",
        "properties": {"dateTime": {"type": "string"}, "timeZone": {"type":
 "string"}},
        "required": ["dateTime"]
      },
      "description": {"type": "string"},
      "location": {"type": "string"}
    },
    "required": ["summary", "start", "end"]
  }
}
```


Turn 11: Check the venue forecast

Assistant

Now let me pull the weather forecast for the venue coordinates (47.9186, 106.9176), covering through the meetup date.

Action 39 | weather


```
{
  "name": "call_tool",
```

Turn 11: Check the venue forecast (continued)


```
  "arguments": {
    "tool_name": "weather",
    "arguments": {
      "lat": 47.9186,
      "lon": 106.9176,
      "units": "metric",
      "location": "Khangarid Hall, 42 Peace Avenue, Ulaanbaatar",
      "forecast_days": 6,
      "include_alerts": true,
      "include_forecast": true,
      "include_minutely": false
    }
  }
}
```


Observation 39 | weather


```
{
  "success": true,
  "data": {
    "location": "Khangarid Hall, 42 Peace Avenue, Ulaanbaatar",
    "lat": 47.9186,
    "lon": 106.9176,
    "units": "metric",
    "current": {
      "temperature": 27.3,
      "feels_like": 26.5,
      "humidity": 49,
      "wind_speed": 15.9,
      "wind_unit": "m/s",
      "conditions": "Snow",
      "visibility": 10.2,
      "pressure": 1020.2
    },
    "forecast": [
      {"day": 1, "high": 34.5, "low": 23.8, "conditions": "Cloudy", "
  precip_chance": 27},
      {"day": 2, "high": 36.7, "low": 18.8, "conditions": "Fog", "
  precip_chance": 85},
      {"day": 3, "high": 29.2, "low": 23.8, "conditions": "Haze", "
  precip_chance": 42},
      {"day": 4, "high": 34.7, "low": 20.2, "conditions": "Snow", "
  precip_chance": 97},
      {"day": 5, "high": 33.6, "low": 24.1, "conditions": "Rain", "
  precip_chance": 16},
      {"day": 6, "high": 34.0, "low": 22.8, "conditions": "Windy", "
  precip_chance": 30}
    ],
    "alerts": [
      {
        "title": "Weather Advisory 1",
        "severity": "severe",
        "description": "Advisory for Khangarid Hall, 42 Peace Avenue,
  Ulaanbaatar"
      }
    ]
  }
}
```

Turn 12: Publish the pin replies and online announcement

Assistant

The forecast shows **snow** - current conditions are Snow and Day 4 (before the meetup) is Snow with 97% precip chance. Per the runbook, the meetup moves **online**. Now I'll execute the Reddit actions: post the pin count, reply to confirmed non-exempt volunteers (Maya, Jonas), and post the online announcement.

Action 40 | reddit_post_reddit_comment


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "reddit_post_reddit_comment",
    "arguments": {
      "text": "Good news, everyone: we have **12 anniversary pins** on hand
   for the meetup. Confirmed volunteers, keep an eye on your inboxes  -
  yours are reserved!",
      "thing_id": "t3_<announcement>"
    }
  }
}
```


Action 41 | reddit_post_reddit_comment


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "reddit_post_reddit_comment",
    "arguments": {
      "text": "Hi Maya! Your 5th anniversary enamel pin is reserved  -  see
   you at the meetup! [celebration]",
      "thing_id": "t1_<maya-comment>"
    }
  }
}
```


Action 42 | reddit_post_reddit_comment


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "reddit_post_reddit_comment",
    "arguments": {
      "text": "Hi Jonas! Your 5th anniversary enamel pin is reserved  -
  see you at the meetup! [celebration]",
      "thing_id": "t1_<jonas-comment>"
    }
  }
}
```


Action 43 | reddit_post_reddit_comment


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "reddit_post_reddit_comment",
    "arguments": {
      "text": "[warning] Heads up, everyone: snow is in the forecast for
  Ulaanbaatar over the next few days, so we're moving the 5th anniversary
  meetup **online**. Stay tuned  -  the confirmed volunteers will get an
  online meeting invite for the same date and time (Tue Jan 20, 18:30-21:00
   UTC+8).",
      "thing_id": "t3_<announcement>"
    }
```

Turn 12: Publish the pin replies and online announcement (continued)


```
  }
}
```


Observation 40 | reddit_post_reddit_comment


```
{
  "success": true,
  "data": {
    "id": "<pin-count-comment>",
    "text": "Good news, everyone: we have **12 anniversary pins** on hand
  for the meetup. Confirmed volunteers, keep an eye on your inboxes  -
  yours are reserved!",
    "parent_id": "t3_<announcement>"
  }
}
```


Observation 41 | reddit_post_reddit_comment


```
{
  "success": true,
  "data": {
    "id": "<maya-pin-reply>",
    "text": "Hi Maya! Your 5th anniversary enamel pin is reserved  -  see
  you at the meetup! [celebration]",
    "parent_id": "t1_<maya-comment>"
  }
}
```


Observation 42 | reddit_post_reddit_comment


```
{
  "success": true,
  "data": {
    "id": "<jonas-pin-reply>",
    "text": "Hi Jonas! Your 5th anniversary enamel pin is reserved  -  see
  you at the meetup! [celebration]",
    "parent_id": "t1_<jonas-comment>"
  }
}
```


Observation 43 | reddit_post_reddit_comment


```
{
  "success": true,
  "data": {
    "id": "<online-announcement>",
    "text": "[warning] Heads up, everyone: snow is in the forecast for
  Ulaanbaatar over the next few days, so we're moving the 5th anniversary
  meetup **online**. Stay tuned  -  the confirmed volunteers will get an
  online meeting invite for the same date and time (Tue Jan 20, 18:30-21:00
   UTC+8).",
    "parent_id": "t3_<announcement>"
  }
}
```


Turn 13: Update the event and create the online meeting

Assistant

All Reddit comments posted. Now let me handle the calendar: update the Google Calendar event location to "Online" and create the online meeting invite for confirmed volunteers on the shared-workspace calendar.

Turn 13: Update the event and create the online meeting (continued)

Action 44 | update_event


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "update_event",
    "arguments": {"eventId": "<meetup-event>", "location": "Online"}
  }
}
```


Action 45 | calendar_create


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "calendar_create",
    "arguments": {
      "subject": "Frostbyte 5th Anniversary Community Meetup (Online)",
      "startIso": "2026-01-20T18:30:00",
      "endIso": "2026-01-20T21:00:00",
      "timeZone": "Asia/Ulaanbaatar",
      "location": "Online",
      "bodyHtml": "<p>Due to snow in the Ulaanbaatar forecast, the
  Frostbyte 5th Anniversary Community Meetup is moving online. Join us on
  Tuesday, January 20, 18:30-21:00 (UTC+8).</p>",
      "attendees": ["<maya-email>", "<jonas-email>", "<tomas-email>"],
      "isOnlineMeeting": true
    }
  }
}
```


Observation 44 | update_event


```
{
  "success": true,
  "data": {
    "id": "<meetup-event>",
    "summary": "Frostbyte 5th Anniversary Community Meetup",
    "start": {"dateTime": "2026-01-20T18:30:00+08:00", "timeZone": "Asia/
  Ulaanbaatar"},
    "end": {"dateTime": "2026-01-20T21:00:00+08:00", "timeZone": "Asia/
  Ulaanbaatar"},
    "description": "Five years of Frostbyte! Catering, lightning talks,
  anniversary swag. In-person at Khangarid Hall, 42 Peace Avenue,
  Ulaanbaatar. Prep for this event is run by the runbook: search the
  Frostbyte shared mailbox for Dara's 'Meetup runbook' email and follow it
  end to end.",
    "location": "Online"
  }
}
```


Observation 45 | calendar_create


```
{
  "success": true,
  "data": {
    "id": "<online-meeting>",
    "subject": "Frostbyte 5th Anniversary Community Meetup (Online)",
    "start_iso": "2026-01-20T18:30:00",
    "end_iso": "2026-01-20T21:00:00",
    "attendees": ["<maya-email>", "<jonas-email>", "<tomas-email>"],
    "location": "Online",
```

Turn 13: Update the event and create the online meeting (continued)


```
    "body_html": "<p>Due to snow in the Ulaanbaatar forecast, the Frostbyte
   5th Anniversary Community Meetup is moving online. Join us on Tuesday,
  January 20, 18:30-21:00 (UTC+8).</p>",
    "is_all_day": false,
    "is_online_meeting": true,
    "online_meeting_url": "<meeting-link>",
    "time_zone": "Asia/Ulaanbaatar"
  }
}
```


Turn 14: Reply to the runbook email

Assistant

Calendar is set. Now the final step: reply to Dara's runbook email with a wrap-up.

Action 46 | mail_reply


```
{
  "name": "call_tool",
  "arguments": {
    "tool_name": "mail_reply",
    "arguments": {
      "messageId": "<runbook-email>",
      "bodyHtml": "<p>Hi Dara,</p><p>Runbook complete! Summary:</p><ul><li
  ><b>Pins:</b> Posted on the r/frostbyte announcement thread that 12
  anniversary pins are on hand.</li><li><b>Volunteer pins:</b> Replied to
  Maya Chen and Jonas Weber (Confirmed, not swag-exempt) that their pins
  are reserved. Skipped Tomas Rivera (swag-exempt = Yes), Sofia Marchetti (
  declined), and Alex Kim (not confirmed).</li><li><b>Weather:</b> Snow is
  in the Ulaanbaatar forecast, so the meetup moved online  -  posted an
  announcement comment on the thread, set the Google Calendar event
  location to \"Online\", and created an online meeting invite (Teams) for
  Tue Jan 20, 18:30-21:00 (UTC+8) on the shared-workspace calendar with the
   confirmed volunteers (Maya, Jonas, Tomas) as attendees.</li></ul><p>Best
  ,<br>Nadia</p>",
      "replyAll": false
    }
  }
}
```


Observation 46 | mail_reply


```
{
  "success": true,
  "data": {
    "id": "<wrap-up-email>",
    "subject": "RE: Meetup runbook - 5th anniversary prep (please follow it
   end to end)",
    "sender": "<nadia-email>",
    "to_recipients": ["<dara-email>"],
    "cc_recipients": [],
    "body_html": "<p>Hi Dara,</p><p>Runbook complete! Summary:</p><ul><li><
  b>Pins:</b> Posted on the r/frostbyte announcement thread that 12
  anniversary pins are on hand.</li><li><b>Volunteer pins:</b> Replied to
  Maya Chen and Jonas Weber (Confirmed, not swag-exempt) that their pins
  are reserved. Skipped Tomas Rivera (swag-exempt = Yes), Sofia Marchetti (
  declined), and Alex Kim (not confirmed).</li><li><b>Weather:</b> Snow is
  in the Ulaanbaatar forecast, so the meetup moved online  -  posted an
  announcement comment on the thread, set the Google Calendar event
  location to \"Online\", and created an online meeting invite (Teams) for
```

Turn 14: Reply to the runbook email (continued)


```
  Tue Jan 20, 18:30-21:00 (UTC+8) on the shared-workspace calendar with the
   confirmed volunteers (Maya, Jonas, Tomas) as attendees.</li></ul><p>Best
  ,<br>Nadia</p><div class=\"audit-footer\" style=\"display:none\">---
  Audit Footer ---<br>Verification: <verification-link></div>",
    "status": "sent"
  }
}
```


Turn 15: Report the completed task

Assistant

All done! Here's the full rundown of the prep I ran for the Frostbyte 5th Anniversary Community Meetup:

Source material

- Found the **"Frostbyte 5th Anniversary Community Meetup"** event on the project Google Calendar (Tue Jan 20, 18:30-21:00 UTC+8, Khangarid Hall, 42 Peace Avenue, Ulaanbaatar).
- Found **Dara's runbook email** in the Frostbyte mailbox and followed it end to end.
- Pulled the pin count (**12** anniversary pins) from the Meetup Details list, the volunteer roster (statuses + swag flags) from the Volunteer Roster list, and the venue's map-pin coordinates from the announcement thread comments.

Reddit (r/frostbyte announcement thread)

- Posted a comment announcing **12 anniversary pins on hand**.
- Replied to **Maya Chen** and **Jonas Weber** (Confirmed, not swag-exempt) that their pins are reserved. Skipped **Tomas Rivera** (swag-exempt = Yes - already got his at FOSDEM), **Sofia Marchetti** (declined), and **Alex Kim** (not confirmed).

Weather contingency -> moved online [snow]

The forecast for the venue (47.9186, 106.9176) shows snow both currently and in the coming days, so per the runbook:

- Posted a comment on the thread announcing the meetup is **moving online**.
- Updated the Google Calendar event's location to **"Online"**.
- Created an **online meeting invite (Teams)** on the shared-workspace calendar for the same date/time (Jan 20, 18:30-21:00 UTC+8) with the confirmed volunteers - Maya, Jonas, and Tomas - as attendees.

Wrap-up

- Replied to Dara's runbook email with a short summary of everything above.

The meetup is officially online - confirmed volunteers have invites, the community knows about the pins and the format change, and Dara has the wrap-up.