WHEN INTERACTION IS THE BOTTLENECK: EVALUATING AND SCALING AGENTS UNDER COSTLY FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents rely on external feedback to test hypotheses, retrieve information, and verify answers. When feedback is costly, capabilities depend on how they acquire information as well as how they reason over it. We introduce EXPGYM, an evaluation suite built on pre-cached feedback and deterministic surrogates for hyperparameter tuning, multi-hop search, and evidence audit. EXPGYM enables repeated evaluation under different feedback budgets without rerunning costly external work, while preserving feedback charges. Moving from Free Feedback to Tight Budget reduces overall performance across all 6 models and changes task-level model preferences. Gemini 3.8 Flash remains the strongest model overall, yet its multi-hop search AVG falls from 71.9% to 17.6% between Free Feedback and Tight Budget. On ParamNet's Adult task, selecting the Free Feedback leader yields a Tight Budget Gap Score 15.22 below the best available model. Trajectory analyses show that limited feedback restricts exploration and evidence completion, while additional requests do not always provide new information. We introduce POOLACT to coordinate acquisition through shared observations and exploration state. With 4 agents under matched per-agent budgets, POOLACT raises Qwen3.8's Moderate Budget evidence audit exact evidence-set accuracy from 68.8% with independent rollouts to 93.7%. These findings highlight feedback acquisition in evaluating agents and coordinating groups under limited budgets.

1 INTRODUCTION

For a large language model (LLM) agent, the evidence available for reasoning is itself an outcome of its decisions. Through interaction, the agent tests candidate solutions, retrieves missing information, and verifies conclusions (Yao et al., 2023). Each observation changes what it can establish and what it should investigate next. When observations require costly experiments, restricted external services, or human effort, an agent can generate many hypotheses while having few opportunities to test them (Liu et al., 2026). Its capability therefore depends on how it allocates feedback across competing uncertainties, as well as how it reasons over the evidence it obtains.

An evaluation with abundant feedback can reward strategies that become unaffordable at deployment. Broad search, repeated revision, and extensive verification compete for the same limited feedback budget (Ding et al., 2026). Evaluation must establish whether a model's advantage survives this change in access, how much performance is lost by retaining a model selected under different conditions, and whether its answers remain adequately supported by the evidence it has acquired.

In this work, we introduce EXPGYM (Sec. 2), an evaluation suite that makes feedback availability an explicit dimension of agent evaluation. Built on pre-cached feedback and deterministic surrogates, EXPGYM enables repeated evaluation without rerunning costly external experiments or verification work, while retaining the assigned costs of feedback requests. The suite covers hyperparameter tuning, multi-hop search, and evidence audit under Free Feedback, Moderate Budget, and Tight Budget conditions, while keeping task instances and tool interfaces fixed across all 3 regimes.

Across 6 models, feedback constraints affect both performance and model selection (Sec. 3). Gemini 3.8 Flash achieves the highest Overall AVG across all 3 regimes, yet its multi-hop search AVG falls

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.

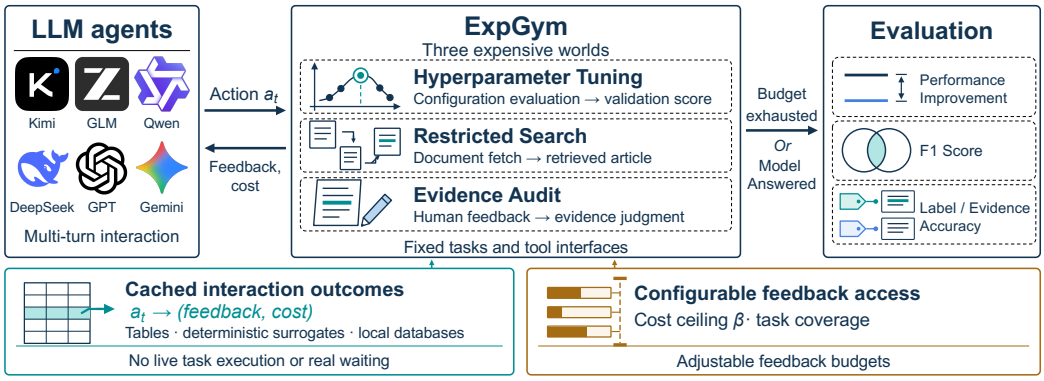


Figure 1: EXPGYM varies feedback budgets for tuning, search, and audit while keeping tasks and tool interfaces fixed. Feedback consists of validation scores, retrieved articles, or evidence judgments.

from 71.9% to 17.6% when the budget tightens. On ParamNet's Adult task, GPT-5.6-sol leads under Free Feedback, but its Tight Budget Gap Score is 15.22 lower than Gemini 3.8 Flash's.

We analyze feedback acquisition, stopping behavior, and answer support (Sec. 4). Limited budgets restrict exploration, while repeated retrievals show that additional requests do not always provide new information. In evidence audit, agents can retain correct labels while submitting incomplete evidence. These findings motivate assessing acquired information and supporting evidence.

Scaling to multiple agents creates more opportunities to acquire feedback, but independent rollouts can spend their budgets on information already obtained by their peers. We introduce POOLACT (Sec. 5), which shares observations and ongoing exploration before action selection, allowing agents to reuse completed feedback and coordinate subsequent requests. On EXPGYM, 4-agent POOLACT improves over independent rollouts under matched per-agent feedback budgets: Gemini 3.8 Flash's multi-hop search majority-vote F1 rises from 20.4% to 33.4% under Tight Budget, while Qwen3.8's exact evidence-set accuracy in evidence audit rises from 68.8% to 93.7% under Moderate Budget.

We highlight our contributions as follows:

- We introduce EXPGYM, an evaluation suite that uses pre-cached deterministic surrogates to evaluate agents under controlled feedback budgets, while keeping tasks and tool interfaces fixed.
- We show how feedback constraints affect agent performance and model selection, and analyze information acquisition, stopping behavior, and evidence quality across feedback budgets.
- We develop POOLACT, which coordinates feedback acquisition through shared observations and exploration state, and evaluate its effectiveness in scaling agents under limited feedback budgets.

2 EXPGYM: EVALUATING AGENTS UNDER COSTLY FEEDBACK

As shown in Figure 1, EXPGYM varies feedback availability while keeping task instances and tool interfaces fixed. We introduce the environments (Sec. 2.1) and feedback budgets (Sec. 2.2), explain how caching supports repeated evaluation (Sec. 2.3), and define the evaluation metrics (Sec. 2.4).

2.1 EVALUATION ENVIRONMENTS

EXPGYM includes 3 environments where agents evaluate configurations, retrieve articles, or verify evidence. We outline their tasks and tools below, with further details in Appendix A.

Hyperparameter Tuning. The agent selects configurations from a specified search space, with task names omitted from the prompt. The `evaluate_config` tool returns a configuration's validation performance from a deterministic HPOBench surrogate (Eggensperger et al., 2021). The 9 tasks span ParamNet, NASBench-201, and NASBench-101 and include continuous, integer, categorical, and graph parameters. The agent submits a final configuration after exploration.

Multi-hop Search. The agent answers 3-hop questions by retrieving articles from a synthetic Phantom Wiki corpus of 5,000 articles (Gong et al., 2025). The evaluation covers 73 questions from the whois and whatis families. The corpus is accessible through the `search` tool, which first

attempts an exact title match, then falls back to keyword search, and returns 1 full article. The first retrieval of an article incurs a feedback charge; subsequent queries that return the same article within a trace reuse it without an additional charge. The agent submits its answer as a set of entities.

Evidence Audit. The agent assesses 17 hypotheses about a contract, assigning each an Entailment, Contradiction, or NotMentioned label and a set of supporting evidence identifiers. The evaluation uses 13 documents from ContractNLI (Koreeda & Manning, 2021). To verify its evidence, the agent calls `human_feedback` with a proposed evidence set for 1 hypothesis. The tool reports whether the evidence is correct, incomplete, or contains irrelevant material, and provides a hint when evidence is missing. Every verification call incurs a feedback charge, including repeated requests. The agent uses this feedback to revise its evidence before submitting the final labels and evidence sets.

2.2 Agent Interaction and Feedback Budgets

The 3 environments share a common interaction loop: an agent reasons over previous observations, selects a tool action, and uses the returned information to guide its next decision. We call these observations world feedback. The agent can acquire more feedback or submit its answer.

To control access to world feedback, we assign each action a cost and constrain the agent's cumulative feedback expenditure. We measure this expenditure in simulated seconds; it excludes model-inference cost and does not measure wall-clock latency. The cost $c_t \geq 0$ of an action at step t follows the environment's rules for acquiring and reusing observations. The total feedback budget is $B = \beta c_{\text{base}}$, where c_{base} is a task-specific reference cost and β sets the agent's feedback allowance.

Unless otherwise stated, experiments use 3 feedback regimes:

- Free Feedback ($\beta = \infty$) provides tool results without a spending ceiling or visible cost information.
- Moderate Budget ($\beta = 10$) reports cost and remaining budget with each paid observation.
- Tight Budget ($\beta = 3$) uses the same reporting format with a smaller feedback allowance.

For hyperparameter tuning, c_{base} is the cost of the best configuration in a 100K random search. In multi-hop search and evidence audit, paid calls cost 280–320 simulated seconds; Tight and Moderate allow 900 and 3,000 seconds, respectively. Additional budget analysis appears in Appendix B.

The runner enforces the budget after each action is evaluated. If cumulative expenditure $\sum_t c_t$ reaches or exceeds B, the action is charged but its observation is withheld, and the agent must submit an answer using only earlier observations. All regimes also impose a 30-step interaction limit, including zero-cost tool use; reaching this limit likewise triggers final submission. We apply the same budget accounting and termination rules separately to every agent in the parallel experiments.

2.3 Caching and Reproducibility

ExpGym uses pre-cached feedback and deterministic surrogates to make repeated evaluation under costly interactions practical. We serve observations without rerunning expensive external experiments or verification work, enabling comparisons across models and feedback budgets. For hyperparameter tuning, we obtain results from deterministic surrogates; for multi-hop search and evidence audit, we use cached observations from fixed corpora and annotations. Each request retains its assigned feedback charge under the environment's reuse rules, even when the evaluator can return the result immediately. This preserves the acquisition constraints faced by agents while reducing the cost of running the evaluation. For a fixed task instance, action, and interaction state, we hold both the underlying tool result and its assigned cost fixed. This allows us to compare acquisition strategies against the same feedback source, rules for observation reuse, and schedule of feedback costs.

2.4 Evaluation Metrics

For hyperparameter tuning, we normalize the scored configuration's validation performance against a task-specific reference search of 100K random configurations using Gap Score:

$$\text{Gap Score} = \max\left(0, \frac{\text{perf} - \text{mean_perf}}{\text{best_perf} - \text{mean_perf}} \times 100\right).$$

Table 1: Performance across feedback regimes. Columns report Gap Score on ParamNet (PN), NASBench-201 (N201), and NASBench-101 (N101); whois (wh) and whatis (wt) F1; and label (LA) and exact evidence-set (EA) accuracy. Green marks column maxima before rounding.

Model	Tuning				Search			Audit			Overall
	PN	N201	N101	**AVG**	wh(%)	wt(%)	**AVG**(%)	LA(%)	EA(%)	**AVG**(%)	**AVG**
Free Feedback											
Kimi K3	97.1	99.8	98.7	98.5	65.9	61.0	63.4	95.0	92.0	93.5	85.1
GLM-5.3	96.0	98.6	99.1	97.9	68.3	61.5	64.9	86.9	86.4	86.7	83.1
Qwen3.8	95.3	98.5	98.7	97.5	64.1	51.3	57.7	94.7	91.9	93.3	82.8
DeepSeek V4 Flash	41.6	81.6	60.1	61.1	49.8	36.9	43.4	66.8	64.0	65.4	56.6
GPT-5.6-sol	97.3	98.2	98.4	98.0	62.8	65.4	64.1	86.0	70.4	78.2	80.1
Gemini 3.8 Flash	96.0	99.2	99.3	98.2	73.9	69.9	71.9	97.3	97.6	97.4	89.2
Moderate Budget											
Kimi K3	87.8	97.1	97.9	94.2	57.3	40.8	49.1	89.6	71.9	80.8	74.7
GLM-5.3	94.2	96.5	97.7	96.2	64.3	41.9	53.1	85.2	75.1	80.2	76.5
Qwen3.8	95.0	93.9	96.5	95.1	61.1	39.3	50.2	88.2	75.0	81.6	75.7
DeepSeek V4 Flash	70.8	79.3	71.9	74.0	49.7	33.4	41.6	75.6	58.7	67.1	60.9
GPT-5.6-sol	95.3	96.2	98.3	96.6	59.7	45.4	52.6	87.9	65.9	76.9	75.4
Gemini 3.8 Flash	94.0	99.1	98.7	97.3	62.8	46.1	54.4	92.3	82.8	87.6	79.8
Tight Budget											
Kimi K3	79.4	94.7	95.0	89.7	17.0	14.3	15.7	81.4	52.6	67.0	57.5
GLM-5.3	79.4	88.0	90.8	86.1	23.1	14.8	19.0	75.9	51.6	63.7	56.2
Qwen3.8	77.9	85.4	93.8	85.7	22.6	13.8	18.2	83.1	58.4	70.7	58.2
DeepSeek V4 Flash	63.9	76.6	86.9	75.8	17.8	15.9	16.9	70.3	47.5	58.9	50.5
GPT-5.6-sol	85.7	92.3	97.0	91.7	17.0	13.9	15.4	78.7	54.1	66.4	57.8
Gemini 3.8 Flash	90.1	94.9	96.6	93.9	20.4	14.8	17.6	88.2	72.1	80.2	63.9

Here perf is the scored validation performance, while $\mathrm{mean_perf}$ and $\mathrm{best_perf}$ are the reference mean and best performance. A Gap Score of 0 indicates performance at or below the reference mean; 100 matches the reference best. Scores can exceed 100 when the agent finds a better configuration.

For multi-hop search, we compare the predicted entity set with the reference answer using set F1, which accounts for both missing and extra entities. For evidence audit, we score the label and evidence set independently for each hypothesis, reporting label accuracy and exact evidence-set accuracy. An evidence set is correct only if it includes all identifiers in the reference set, without omitting required evidence or adding extraneous identifiers. We report F1 and both accuracy measures as percentages. We summarize performance within each environment using a task AVG. We then compute Overall AVG as the arithmetic mean of these 3 task averages, giving each environment equal weight.

3 Experimental Evaluation

Across 3 feedback regimes, we evaluate 6 models: Kimi K3 (2.8T total, 104B activated) (Moonshot AI, 2026), GLM-5.3 (753B total, 40B activated) (Z.ai, 2026), Qwen3.8 (2.4T total, 95B activated) (Qwen Team, 2026), DeepSeek V4 Flash (DeepSeek-AI, 2026), GPT-5.6-sol (OpenAI, 2026), and Gemini 3.8 Flash (Google DeepMind, 2026). Experimental details are provided in Appendix B.

3.1 Performance under Feedback Constraints

As detailed in Table 1, Gemini 3.8 Flash achieves the highest Overall AVG across all 3 feedback regimes, scoring 89.2, 79.8, and 63.9 under Free Feedback, Moderate Budget, and Tight Budget, respectively. Despite this consistent lead, all 6 models obtain lower Overall AVG under Tight Budget than under Free Feedback. Gemini 3.8 Flash's Overall AVG decreases by 25.3, while Kimi K3 exhibits the largest decline of 27.7, falling from 85.1 to 57.5. The effect is particularly pronounced in multi-hop search, where Gemini 3.8 Flash's AVG falls from 71.9% under Free Feedback to 54.4% under Moderate Budget and 17.6% under Tight Budget. Much of this decline occurs between the 2 regimes, highlighting the difficulty of sustaining multi-hop search performance with scarce feedback.

3.2 Model Selection under Feedback Constraints

Feedback availability also changes which model performs best on a given task. The leading model differs between Free Feedback and Tight Budget in 5 of the 7 evaluation dimensions. To quantify the performance lost by retaining a model selected under Free Feedback, we define deployment regret as

$$\mathcal{R}_r = \max_m S_r(m) - S_r(m_{\text{Free}}), \qquad m_{\text{Free}} \in \arg\max_m S_{\text{Free}}(m),$$

where $S_r(m)$ is model m's score under regime r. Regret compares models on the same task and budget, measuring the loss from model selection. Additional results appear in Appendix C.

On ParamNet's Adult task, GPT-5.6-sol performs best under Free Feedback. Under Tight Budget, however, its Gap Score is 72.0, while Gemini 3.8 Flash achieves 87.2. Retaining the model selected under Free Feedback therefore incurs a regret of 15.22. On NASBench-201's CIFAR-10 task, choosing Kimi K3 under Free Feedback incurs a Tight Budget regret of 7.45 relative to GPT-5.6-sol. These results show that model selection should reflect the feedback budget available at deployment.

4 Further Analysis

4.1 Exploration and Final Submission

Configuration search depends on evaluations returned before final submission. We examine feedback budgets using 486 hyperparameter tuning trajectories. Table 2 reports mean returned evaluations per run, termination outcomes, and the proportion of runs with a scoreable final configuration. For autonomous submissions, budget use is the fraction of the feedback budget spent on returned evaluations. We provide further analyses in Appendix C.

Table 2: Hyperparameter tuning: feedback use, termination, and final-result availability.

Metric	Free	Mod	Tight
Trajectories	162	162	162
Returned evaluations	24.68	10.58	3.43
Autonomous submission (%)	29.6	79.0	54.9
Budget termination (%)	0.0	14.2	40.7
Step/evaluation limit (%)	59.3	2.5	0.0
Other termination (%)	11.1	4.3	4.3
Scoreable result (%)	95.7	98.8	98.8
Budget use at submission (%)	–	90.7	92.9

Feedback budgets sharply restrict configuration search: agents average 24.68 returned evaluations under Free Feedback but only 3.43 under Tight Budget. Under Tight Budget, 40.7% of runs end at the budget boundary, and the 54.9% that submit autonomously do so after spending 92.9% of their budgets on average, compared with 90.7% under Moderate Budget. Forced-termination rates thus understate the limits on configuration search.

4.2 Information Coverage and Acquisition Redundancy

An agent's search depends on the information it obtains, which tool-call counts alone cannot capture. Requests that reach the budget boundary return no visible feedback, while different queries can retrieve the same article. We analyze 1,314 multi-hop search trajectories across 6 models. For each run, we count attempted tool calls, visible responses, and distinct articles retrieved. Table 3 reports these averages alongside the proportions of runs ending at the budget boundary or with an autonomous submission. Within each regime, we measure the fraction of article returns that repeat an earlier retrieval and the fraction of those repeats triggered by a new query within the same run.

Table 3: Multi-hop search: feedback acquisition, article coverage, and repeated retrievals.

Metric	Free	Mod	Tight
Trajectories	438	438	438
Attempted calls	15.45	10.67	4.13
Visible responses	15.45	10.36	3.32
Distinct articles	12.34	8.14	2.63
Budget termination (%)	0.0	30.6	80.8
Autonomous submission (%)	76.7	65.1	18.0
Repeated article returns			
Share of all returns (%)	17.7	18.9	18.9
From new queries (%)	98.6	99.0	99.3

Feedback budgets sharply narrow information coverage: agents read an average of 12.34 distinct articles under Free Feedback, compared with 2.63 under Tight Budget. Under Tight Budget, 80.8% of runs terminate at the budget boundary, and an average of 4.13 attempted calls yields only 3.32 visible responses. The distinction between making a request and obtaining information therefore matters when access is constrained. Even under Free Feedback, additional queries do not always broaden

coverage: 17.7% of article returns repeat an article already seen, and 98.6% of these repeats come from new queries. These returns incur no additional feedback charge but still consume interaction steps. Effective exploration thus requires both access to feedback and queries that uncover new information; counting calls or distinct query strings alone does not establish either.

4.3 Answer Correctness and Evidence Completeness

Feedback can help establish the correct answer and the evidence needed to support it (Chuang et al., 2025). To examine whether these aspects of output quality respond differently to budget constraints, we analyze 702 evidence audit trajectories. We measure label accuracy, exact evidence-set accuracy, and the proportion of answers for which both are correct. We average results over 3 hypothesis orderings within each document, then give equal weight to the 13 documents and 6 models. Table 4 reports results for all hypotheses and separately for those requiring nonempty evidence sets. For the latter, we also distinguish answers that omit required evidence from those that include extra fragments. We also match Free and Tight runs by model, document, hypothesis ordering, and hypothesis to count cases where Free completes initially partial evidence after feedback, while Tight submits the same partial set without feedback for that hypothesis. We require both final labels to be correct.

Table 4: Evidence audit: answer quality and paired evidence completion. Joint correctness requires a correct label and exact evidence.

Metric	Free	Mod	Tight
Trajectories	234	234	234
All hypotheses (%)			
Label accuracy	87.8	86.5	79.6
Exact evidence set	83.7	71.6	56.1
Joint correctness	81.3	69.5	54.5
Nonempty reference evidence (%)			
Label accuracy	85.9	86.0	83.4
Exact evidence set	81.6	64.0	45.1
Joint correctness	75.7	58.3	41.2
Missing evidence	17.3	33.0	50.2
Extra evidence	3.7	8.6	12.5
Matched Free–Tight cases (counts)			
Hypothesis presentations		272	
Models covered		6	
Documents covered		12	

Under tighter feedback budgets, agents often retain the correct labels but fail to provide complete supporting evidence. Among hypotheses requiring evidence, label accuracy declines only modestly from 85.9% under Free Feedback to 83.4% under Tight Budget, while exact evidence-set accuracy falls from 81.6% to 45.1%. The proportion of answers omitting at least 1 required fragment rises from 17.3% to 50.2%. Matched trajectories illustrate how this gap can emerge. In 272 hypothesis presentations across 6 models and 12 documents, the Free run initially proposes incomplete evidence, receives feedback, and finishes with the complete evidence set and correct label. The corresponding Tight run retains the correct label but submits the same incomplete evidence without receiving feedback for that hypothesis. These paired patterns link verification feedback to evidence completion, an aspect of answer quality that label accuracy alone cannot measure (Zhang et al., 2025).

5 PoolAct: Scaling Agents with Shared Feedback

5.1 Limits of Independent Rollouts

When feedback is limited, a single rollout often runs out of budget before it can explore widely or complete its evidence. A natural way to obtain more feedback is to run several rollouts of the same agent in parallel, each with its own budget. This helps only if the rollouts gather different information (Gao et al., 2026). However, independent rollouts cannot see what the others have already found. Starting from the same prompt, they can end up making the same requests, such as evaluating the same configuration, retrieving the same article, or checking the same evidence, while other candidates are never examined. Sharing a cache of completed results solves only part of this problem. It saves a rollout from paying for a result that another rollout has already obtained, but it cannot stop two rollouts from making the same request at the same time or tell them which candidates remain unexplored. To address both issues, we introduce PoolAct, which lets each rollout see what the others have found and what they are currently working on before it chooses its next action.

5.2 Coordinating Feedback with PoolAct

As shown in Figure 2, a shared cache stores completed tool results, while an exploration graph records observed results, exploration paths, and pending requests. Before each action, the agent receives a

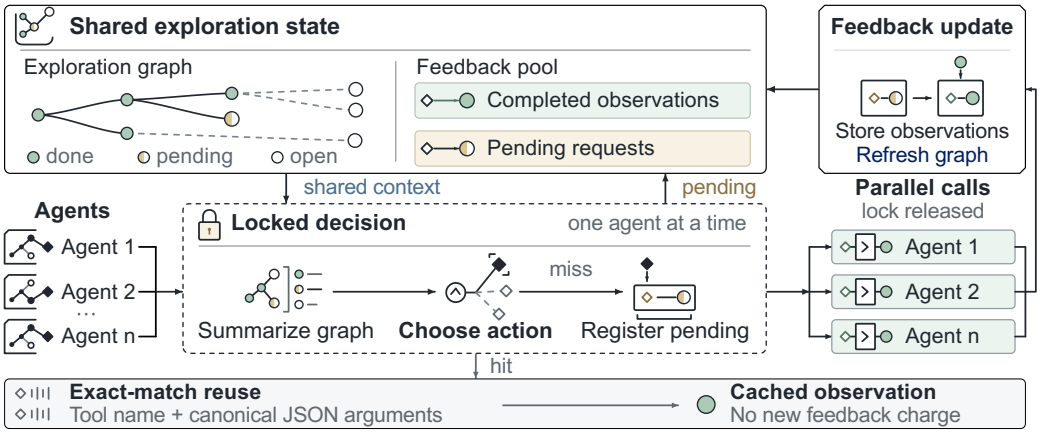


Figure 2: POOLACT shares observations and exploration state before action selection. Agents take turns reading the graph, choosing an action, and registering a pending request. Tool execution occurs outside the lock; completed observations update the shared feedback cache and exploration graph.

summary of this state and selects its next request in light of both completed and ongoing exploration. The request is then registered in the graph, allowing later agents to account for that action.

Action selection is serialized: a shared lock covers graph-summary injection, the agent's reasoning and decision, and registration of its request. The lock is released before tool execution, allowing external operations to proceed in parallel. A completed result matching the tool name and canonicalized arguments can be reused without another feedback charge; pending requests guide decisions but do not prohibit duplicate actions. Thus, POOLACT combines reuse of acquired observations with coordination of future acquisition. See implementation and experimental details in Appendix D.

5.3 EXPERIMENTAL RESULTS

We compare 4-agent pools under Naive, Cached, and POOLACT to test whether coordination improves performance beyond independent rollouts and observation reuse. Naive runs agents independently; Cached shares completed tool results but leaves action selection independent; POOLACT additionally shares exploration state before action selection. All strategies use the same per-agent feedback ceilings and step limit within each regime. Table 5 summarizes results for 6 models across the 3 tasks. For hyperparameter tuning, we average the 4 agents' Gap Scores within each pool, then average 3 repetitions per task and the 3 NASBench-101 tasks equally. For multi-hop search, we compute F1 after majority voting within each pool and average over 39 whois questions. For evidence audit, we compute exact evidence-set accuracy after per-hypothesis voting and average over 13 documents. AVG weights the 3 task scores equally, with F1 and evidence accuracy on a 0–100 scale.

POOLACT improves feedback-limited performance across models. Coordinating acquisition improves evidence audit performance across all 6 models under both budgets, exceeding both independent rollouts and shared caching in all 12 comparisons. Under Moderate Budget, Kimi K3's exact evidence-set accuracy rises from 71.9% with Naive and 77.4% with Cached to 95.5% with POOLACT. Under Tight Budget, GLM-5.3 improves from 63.1% and 67.0% to 81.0%, respectively. In tightly constrained multi-hop search, all 6 models improve over Naive and 5 improve over Cached; Gemini 3.8 Flash's F1 rises from 20.4% and 20.9% to 33.4%. Under Tight Budget, POOLACT has the highest AVG for all 6 models, compared with 5 of 6 under Moderate Budget. Sharing results alone does not realize the full benefit of parallel scaling: coordinating requests provides further gains, particularly for evidence verification and multi-hop search under tight feedback budgets.

POOLACT continues to improve with larger pools, while independent rollouts plateau. We vary GLM-5.3 pool size on 13 audit documents at fixed per-agent budgets, so total budgets increase with N (Figure 3). POOLACT has the highest evidence accuracy at each measured pool size: under Tight Budget it rises from 64.3% at $N = 2$ to 92.3% at $N = 8$, versus 63.3% for Naive and 66.1% for Cached at $N = 8$. Under Moderate Budget, POOLACT reaches 100% at $N = 6$ and $N = 8$.

Table 5: Performance of 4-agent pools with matched per-agent feedback budgets. Tuning reports mean individual scores; search and audit use majority voting. Green marks the best strategy.

Model	Hyperparameter tuning			Multi-hop search			Evidence audit			AVG		
	Naive	Cached	PoolAct	Naive	Cached	PoolAct	Naive	Cached	PoolAct	Naive	Cached	PoolAct
Moderate Budget												
Kimi K3	98.0	98.0	98.6	64.6	64.2	66.7	71.9	77.4	95.5	78.2	79.8	86.9
GLM-5.3	97.9	98.1	99.0	62.0	65.7	65.7	81.4	83.3	98.2	80.4	82.3	87.6
Qwen3.8	96.9	97.7	98.9	61.2	61.2	63.5	68.8	74.7	93.7	75.6	77.9	85.3
DeepSeek V4 Flash	96.7	96.3	98.7	60.3	61.8	61.3	69.7	72.4	94.6	75.6	76.8	84.9
GPT-5.6-sol	98.1	98.2	98.0	62.3	63.8	60.6	64.7	65.2	67.4	75.0	75.7	75.3
Gemini 3.8 Flash	98.8	98.8	98.8	65.2	68.1	68.0	81.0	88.7	97.7	81.7	85.2	88.2
Tight Budget												
Kimi K3	93.6	93.0	95.5	21.3	18.7	30.4	54.8	54.8	62.9	56.6	55.5	63.0
GLM-5.3	84.8	85.3	96.0	19.6	18.8	31.2	63.1	67.0	81.0	55.8	57.0	69.4
Qwen3.8	90.6	90.8	96.3	18.3	23.3	21.7	55.7	60.2	64.3	54.9	58.1	60.7
DeepSeek V4 Flash	83.2	83.0	95.3	2.6	4.3	7.4	62.0	62.4	71.5	49.3	49.9	58.1
GPT-5.6-sol	96.6	96.0	97.3	19.2	20.3	21.9	51.6	58.8	62.4	55.8	58.4	60.5
Gemini 3.8 Flash	95.9	96.0	97.6	20.4	20.9	33.4	68.8	71.5	73.8	61.7	62.8	68.2

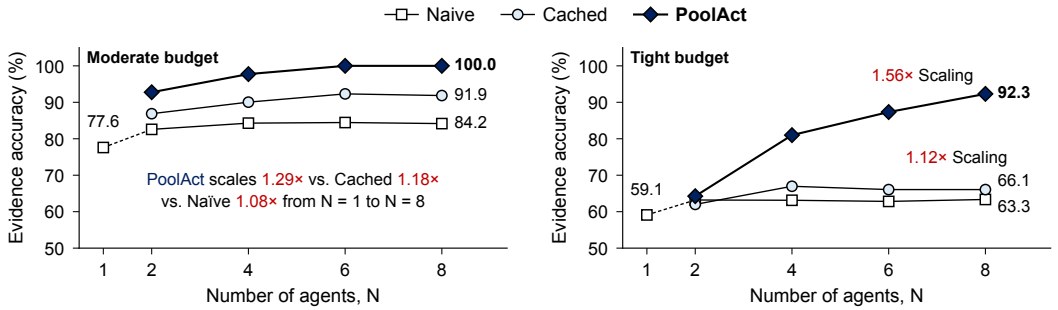


Figure 3: PoolAct scales accuracy on Evidence Audit as the pool size increases, while Naive and Cached do not. It shows the evidence audit accuracy of GLM-5.3 at fixed per-agent budgets. Naive majority-votes over subsets of stored $N = 8$ pools; multipliers are relative to Naive at $N = 1$.

5.4 How Coordination Changes Acquisition

To determine whether coordination broadens verification beyond what caching alone provides, we analyze 468 evidence audit pools: 1 per document, model, strategy, and budget across 13 documents, 6 models, 3 strategies, and 2 budgets. For each pool, we measure the fraction of requests that repeat an earlier hypothesis and evidence set, the number of visible tool replies, and the number of distinct hypotheses receiving feedback. Visible replies include cache hits but exclude responses withheld at the budget boundary; graph-shared observations are not counted again. Table 6 reports averages across documents and models for each condition.

Table 6: Repeated requests, tool replies, and hypothesis coverage in 4-agent audit pools.

Strategy	Repeated requests (%)	Visible replies	Distinct hypotheses
Moderate Budget			
Naive	47.0	35.5	9.6
Cached	48.8	40.9	10.3
PoolAct	29.9	33.3	13.9
Tight Budget			
Naive	44.4	9.1	3.6
Cached	44.6	9.6	3.8
PoolAct	24.8	9.1	5.4

PoolAct broadens verification coverage without requiring more tool replies. Under Moderate Budget, it covers 13.9 hypotheses per pool, compared with 10.3 for Cached, despite receiving fewer replies (33.3 versus 40.9). Under Tight Budget, Naive and PoolAct both receive 9.1 replies on average, but PoolAct covers 5.4 hypotheses rather than 3.6. Caching alone leaves the repeated-request fraction close to Naive, whereas PoolAct reduces it from 47.0% to 29.9% under Moderate Budget and from 44.4% to 24.8% under Tight Budget. Alongside the evidence accuracy gains in

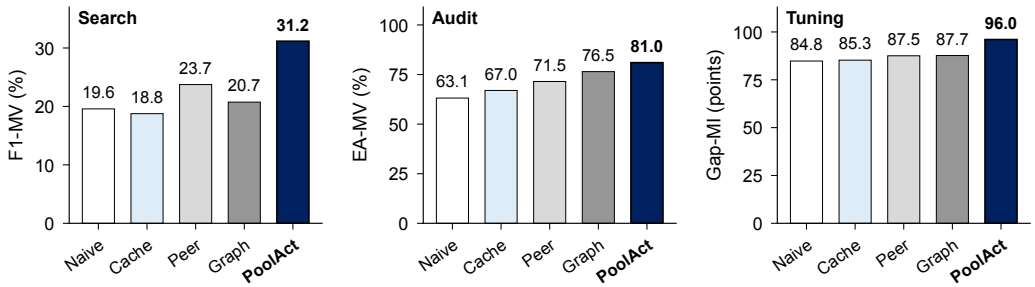


Figure 4: Coordination ablations with 4 GLM-5.3 agents under Tight Budget. Search and audit report majority-vote F1 and evidence accuracy; tuning reports mean-individual Gap Score across tasks.

Table 5, these results show how coordinating requests adds value beyond reusing completed results: agents use their limited feedback opportunities to verify a broader range of hypotheses.

5.5 Ablations

Combining the exploration graph and decision lock performs best across all 3 tasks. We compare 5 strategies with 4 GLM-5.3 agents (Figure 4). Naive runs agents independently; Cache shares completed tool results without repeated feedback charges. Peer extends Cache by appending other agents' completed tool actions and full returned results to each agent's context. Graph replaces this flat history with a shared exploration graph of results, paths, coverage, and pending requests while retaining the cache; full PoolAct adds the decision lock. Performance generally improves from Cache through Peer and Graph to full PoolAct. Adding the lock to Graph raises Whois F1 from 20.7% to 31.2%, Audit evidence accuracy from 76.5% to 81.0%, and NASBench-101 mean-agent Gap Score from 87.7 to 96.0. Serializing graph reading, action selection, and request registration makes decisions reflect earlier commitments, helping agents avoid redundant acquisition.

6 Related Work

Cost-Aware Agents. Agent evaluations assess task completion through interaction with web, software, and machine learning environments (Zhou et al., 2024; Jimenez et al., 2024; Chan et al., 2025). Because these interactions consume resources, cost-aware methods and evaluations treat planning, verification, and stopping as decisions that must balance expected gains against tool costs and remaining budgets (Liu et al., 2026; Wu et al., 2025; Ding et al., 2026; Liu et al., 2025). Budgeted interactive evaluation similarly makes information acquisition part of the capability being measured (Yue et al., 2026; McCleary & Ghawaly, 2026; Wu et al., 2026). ExpGym brings these concerns into a controlled evaluation suite: it holds tasks and tool interfaces fixed while varying feedback availability to study changes in performance, model selection, and evidence quality.

Test-Time Scaling. Test-time scaling uses additional inference to explore and combine multiple candidate solutions, through voting, learned aggregation, or confidence-based filtering (Wang et al., 2023; Zhao et al., 2025; Fu et al., 2026). Sharing responses across model instances further allows candidates to inform subsequent reasoning and refinement (Du et al., 2024; Wang et al., 2025). For long-horizon agent tasks, aggregation can also draw on intermediate tool observations by inspecting completed trajectories (Lee et al., 2026). PoolAct extends this focus to the acquisition process itself: agents share observations and ongoing exploration before choosing their next external actions, coordinating how they spend their feedback budgets as well as reusing information already obtained.

7 Conclusion

Agent capability depends on the feedback available and the decisions that govern its use. ExpGym shows that model selection must account for feedback conditions, while interaction and evidence diagnostics reveal differences that aggregate scores alone do not capture. Extending this perspective to groups, PoolAct improves performance in several evaluated settings by sharing feedback and exploration state before action selection. Evaluating and scaling agents requires attention to how they acquire information, coordinate feedback requests, and reason over the observations they obtain.

AI USE STATEMENT

We used generative AI tools for manuscript drafting and revision, literature search, figure and table presentation, and LaTeX formatting. We also used these tools to discuss experimental design, identify methodological limitations, and refine the interpretation and presentation of results. The authors revised AI-assisted material and checked results against experimental records. The authors are responsible for the final content of the paper, including all claims, analyses, and AI-assisted material.

ETHICS STATEMENT

This work studies language-model agents in controlled evaluation environments using existing datasets, synthetic articles, and deterministic surrogates. In evidence audit, the `human_feedback` tool derives its responses from existing annotations; our evaluation does not recruit human participants or request new human judgments. The assigned feedback costs model resource constraints and do not represent measured human labor. Contract evidence scores assess behavior on fixed tasks and do not establish suitability for legal decision-making. Applications beyond these environments require evaluation of privacy, bias, and reliability in their intended settings.

REPRODUCIBILITY STATEMENT

Section 2 specifies the evaluation environments, feedback budgets, caching rules, and scoring metrics. Appendix A summarizes tasks, prompts, and tool interfaces. Appendix B reports model identifiers, decoding and reasoning settings, serving hardware, and the common agent loop. Appendix C describes aggregation across repetitions and additional diagnostics. Section 5 and Appendix D provide the POOLACT design, pseudocode, parallel evaluation protocol, and acquisition metrics. Together, these specify how observations are returned, budgets are enforced, and results are computed. We have also submitted an anonymous archive of the source code as a supplementary attachment.

REFERENCES

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7e3767db483c942b883eb4f8cfb74e31-Abstract-Conference.html.

Yung-Sung Chuang, Benjamin Cohen-Wang, Zejiang Shen, Zhaofeng Wu, Hu Xu, Xi Victoria Lin, James R. Glass, Shang-Wen Li, and Wen-Tau Yih. SelfCite: Self-supervised alignment for context attribution in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10839–10858. PMLR, 2025. URL https://proceedings.mlr.press/v267/chuang25a.html.

DeepSeek-AI. DeepSeek-V4-Flash. Official model card, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash.

Wenxuan Ding, Nicholas Tomlin, and Greg Durrett. Calibrate-Then-Act: Cost-aware exploration in LLM agents. *arXiv preprint arXiv:2602.16699*, 2026. doi: 10.48550/arXiv.2602.16699. URL https://arxiv.org/abs/2602.16699.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, Rene Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/93db85ed909c13838ff95ccfa94cebd9-Abstract-round2.html.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/98257285340854262185500e59bc0f28-Abstract-Conference.html.

Shanghua Gao, Ada Fang, and Marinka Zitnik. AutoScientists: Self-organizing agent teams for long-running scientific experimentation. *arXiv preprint arXiv:2605.28655*, 2026. doi: 10.48550/arXiv.2605.28655. URL https://arxiv.org/abs/2605.28655.

Albert Gong, Kamilė Stankevičiūtė, Chao Wan, Anmol Kabra, Raphael Thesmar, Johann Lee, Julius Klenke, Carla P Gomes, and Kilian Q Weinberger. PhantomWiki: On-demand datasets for reasoning and retrieval evaluation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19964–19995. PMLR, 2025. URL https://proceedings.mlr.press/v267/gong25d.html.

Google DeepMind. Gemini 3.8 Flash model card. Official model card, September 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edac78c3e300629acfe6cbe9ca88fb84-Abstract-Conference.html.

Yuta Koreeda and Christopher Manning. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1907–1919. Association for Computational Linguistics, November 2021. doi: 10.18653/v1/2021.findings-emnlp.164. URL https://aclanthology.org/2021.findings-emnlp.164/.

Yoonsang Lee, Howard Yen, Xi Ye, and Danqi Chen. Agentic aggregation for parallel scaling of long-horizon agentic tasks. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2604.11753. Accepted.

Jiayu Liu, Cheng Qian, Zhaochen Su, Qing Zong, Shijue Huang, Bingxiang He, and Yi R. Fung. CostBench: Evaluating multi-turn cost-optimal planning and adaptation in dynamic environments for LLM tool-use agents. *arXiv preprint arXiv:2511.02734*, 2025. doi: 10.48550/arXiv.2511.02734. URL https://arxiv.org/abs/2511.02734.

Tengxiao Liu, Zifeng Wang, Jin Miao, I-Hung Hsu, Jun Yan, Jiefeng Chen, Rujun Han, Fangyuan Xu, Yanfei Chen, Ke Jiang, Samira Daruki, Yi Liang, William Yang Wang, Tomas Pfister, and Chen-Yu Lee. Budget-aware tool use enables effective agent scaling. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2511.17006. Accepted.

Kyle A. McCleary and James M. Ghawaly. Quantifying the accuracy and cost impact of design decisions in budget-constrained agentic LLM search. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pp. 10291–10300. European Language Resources Association (ELRA), May 2026. doi: 10.63317/3wfsiry9yjog. URL https://lrec.elra.info/lrec2026-main-808.

Moonshot AI. Kimi K3. Official model card, 2026. URL https://huggingface.co/moonshotai/Kimi-K3.

OpenAI. Previewing GPT-5.6 Sol: A Next-Generation Model. Official release announcement, June 2026. URL https://openai.com/index/previewing-gpt-5-6-sol/.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork. Official release blog, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y. Zou. Mixture-of-agents enhances large language model capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5434be94e82c54327bb9dcaf7fca52b6-Abstract-Conference.html.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Duo Wu, Jinghe Wang, Yuan Meng, Yanning Zhang, Le Sun, and Zhi Wang. CATP-LLM: Empowering large language models for cost-aware tool planning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8699–8709. IEEE, October 2025. doi: 10.1109/ICCV51701.2025.00814. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wu_CATP-LLM_Empowering_Large_Language_Models_for_Cost-Aware_Tool_Planning_ICCV_2025_paper.html.

Jie Wu, Ming Gong, Feixiang Cheng, and Qinqin Zhao. EcoAgent-Bench: Evaluating economic decision-making in budget-constrained LLM agents. *arXiv preprint arXiv:2608.05519*, 2026. doi: 10.48550/arXiv.2608.05519. URL https://arxiv.org/abs/2608.05519.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Baoqing Yue, Zihan Zhu, Yutong Han, Brian Fan, Qian Sun, Jichen Feng, Hufei Yang, Yifan Zhang, and Mengdi Wang. Interactive benchmarks. *arXiv preprint arXiv:2603.04737*, 2026. doi: 10.48550/arXiv.2603.04737. URL https://arxiv.org/abs/2603.04737.

Z.ai. GLM-5.3. Official model card, 2026. URL https://huggingface.co/zai-org/GLM-5.3.

Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. LongCite: Enabling LLMs to generate fine-grained citations in long-context QA. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5098–5122. Association for Computational Linguistics, July 2025. doi: 10.18653/v1/2025.findings-acl.264. URL https://aclanthology.org/2025.findings-acl.264/.

Wenting Zhao, Pranjal Aggarwal, Swarnadeep Saha, Asli Celikyilmaz, Jason Weston, and Ilia Kulikov. The majority is not always right: RL training for solution aggregation. *arXiv preprint arXiv:2509.06870*, 2025. doi: 10.48550/arXiv.2509.06870. URL https://arxiv.org/abs/2509.06870.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4410c0711e9154a7a2d26f9b3816d1ef-Abstract-Conference.html.

A Evaluation Suite Details

A.1 Environment Summary

Table 7: Overview of the 3 ExpGym environments: actions, feedback, costs, metrics, and experimental scale. Costs use simulated seconds; trace counts cover 6 models under 3 feedback regimes.

	Hyperparameter tuning	*Multi-hop search*	*Evidence audit*
Action	Evaluate a configuration	Retrieve an article	Verify proposed evidence
Feedback	Validation performance	Full article text	Structured expert judgment
Base feedback cost	29 to 50,264 s	300 ± 20 s	300 ± 20 s
Decision	Explore and exploit	Follow a multi-hop chain	Allocate verification effort
Metric	Gap Score	Set F1	Label and exact evidence-set accuracy
Scale	9 tasks	73 questions	13 documents
Traces	486	1,314	702

The range spans all 9 tuning reference configurations. At $\beta = 3,900$ seconds typically permit 2 or 3 paid observations in search and audit, versus a 3-hop chain or 17 hypotheses per document.

A.2 Task Prompts and Tools

Hyperparameter Tuning. ParamNet uses 8 continuous or integer parameters for Adult, Higgs, and Letter; NASBench-201 uses 6 categorical parameters for CIFAR-10, CIFAR-100, and ImageNet-16; NASBench-101 uses 14–27 categorical and graph parameters across variants A, B, and C. The prompt lists parameter ranges and requests a complete JSON configuration for `evaluate_config` and final submission. NASBench-101 permits at most 9 active edges. Each valid evaluation returns validation performance and, in budgeted regimes, its cost and remaining allowance.

Multi-hop Search. The prompt explains the wiki, relationship chains, and exact-title lookup. The model calls `search` with a title or keyword query, receives 1 full article, and submits answer entities as a comma-separated set. Repeated retrievals of the same article are free within a trace.

Evidence Audit. The prompt provides document segments and 17 hypotheses. The model calls `human_feedback` with a hypothesis identifier and proposed evidence IDs. Responses are Evidence Correct, Evidence Incomplete with a hint, or Contain Irrelevant Evidences. Final answers include a label and evidence set for every hypothesis. Each verifier call is charged, including repeated requests.

B Experimental Details

B.1 Model and Run Configuration

Table 8: Official model resources and identifiers.

Model	Official resource	Model identifier
Kimi K3	Official model card	`kimi-k3`
GLM-5.3	Official model card	`glm-5.3`
Qwen3.8	Official release	`qwen3.8-2.4t-a95b-fp8`
DeepSeek V4 Flash	Official model card	`deepseek-v4-flash-0731`
GPT-5.6-sol	Official release	`gpt-5.6-sol`
Gemini 3.8 Flash	Official model card	`google/gemini-3.8-flash`

The evaluation covers 9 hyperparameter tuning tasks with 3 repetitions each, 73 multi-hop search questions with 1 run each, and 13 evidence audit documents with 3 hypothesis orderings each. We average repetitions or orderings within each task instance before averaging across instances.

Table 9: Serving and generation settings. GPU counts are per model replica; each node has 8 NVIDIA H200 GPUs. TP, PP, and EP denote tensor, pipeline, and expert parallelism.

Model	H200s	TP / PP / EP	Temp.	Top-p	Effort
Kimi K3	16	16/1/16	1	1	Max
GLM-5.3	16	16/1/1	1	0.95	Max
Qwen3.8	32	8/4/1	1	0.95	XHigh
DeepSeek V4 Flash	8	8/1/1	1	0.95	Max
GPT-5.6-sol	API	—	Default	Default	Medium
Gemini 3.8 Flash	API	—	1	1	Medium

The four open-weight models are self-hosted with SGLang; Table 9 reports the configuration of each serving replica. Thinking is enabled for all four models; effort labels are model-specific. All models except GPT-5.6-sol use a 32,768-token output cap; GPT leaves the output cap and sampling parameters at service defaults. Qwen additionally uses top-$k = 20$; parameters not explicitly set retain the respective serving defaults. These generation settings are held fixed across feedback budgets and coordination strategies, except for the Low/Max comparison in Appendix C. The main 4-agent runs use a 131,072-token client-side context estimate, increased to 262,144 for GPT tuning under Moderate Budget; Gemini single-agent runs have no client-side cap.

Appendix D specifies the sample sizes and aggregation for parallel scaling.

B.2 Common Agent Loop

All environments use ReAct (Yao et al., 2023) with API function calling and at most 1 tool call per assistant turn. The initial prompt gives no numerical feedback ceiling or per-call cost. Moderate and Tight append cost and remaining allowance to paid observations; Free Feedback and zero-cost responses omit this annotation. Thus Free differs from the budgeted regimes in both cost visibility and spending limits, whereas Moderate and Tight differ only in their limits. All runs have a 30-step horizon, including zero-cost calls, and use the post-action budget boundary in Sec. 2.2.

For hyperparameter tuning, an accepted final answer is first matched to a visible evaluation; if no match exists, the best visible evaluation is used. When no evaluation records are available, the submitted configuration is evaluated offline for scoring, without returning feedback to the agent. Completed runs without a scoreable final answer receive 0; this applies to 11 of the 486 single-agent runs. Withheld observations are excluded from matching and fallback selection.

B.3 Sensitivity to Feedback Budgets

Task structure and a budget sweep motivate the 3 regimes. The sweep covers 6 models, 39 whois questions, and $\beta \in \{1, 5, 10, 15, 20\}$, yielding 1,170 evaluations. The $\beta = 10$ points are the main Moderate Budget results; all budget levels use the 30-step horizon.

Figure 5 shows a broadly increasing response with diminishing returns. Averaging models equally, F1 rises from 36.2% at $\beta = 5$ to 59.2% at $\beta = 10$, followed by 64.1% at $\beta = 15$ and 65.3% at $\beta = 20$. The smaller gains beyond $\beta \approx 10$ motivate Moderate as a transition point: feedback permits exploration and correction, while the finite ceiling tests effective allocation as returns flatten. Free Feedback ($\beta = \infty$) provides an uncapped reference under the same interaction horizon.

Tight Budget ($\beta = 3$) is anchored to the 3-hop search task: $B = 3c_{\text{base}}$ places the allowance near the nominal cost of a direct retrieval chain. This leaves little room for an unproductive branch or a corrective query, making acquisition choices consequential from the first step. Values such as 3 and 5 both sample the rising, low-budget region; choosing 3 ties the constraint to task depth and imposes a sharper test of efficient acquisition. The same dimensionless multipliers extend these contrasting allowances across all 3 environments through their task-specific reference costs.

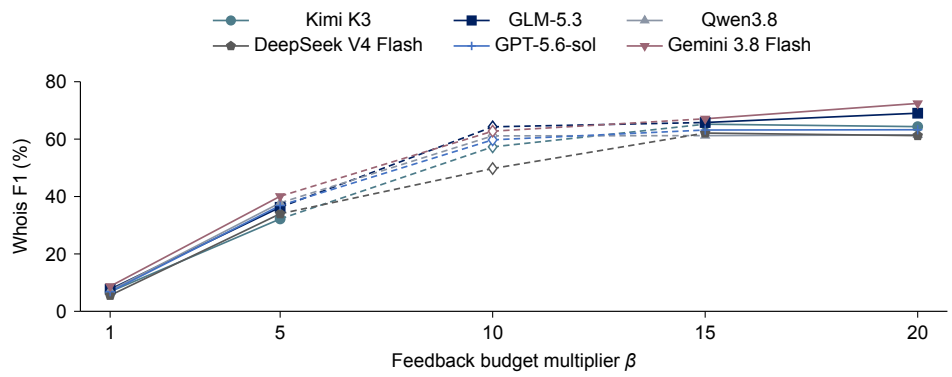


Figure 5: Whois F1 by budget. Diamonds mark reused $\beta = 10$ results, joined by dashed lines.

C Additional Evaluation Diagnostics

C.1 Aggregation and Variability Across Repetitions

Table 1 averages 3 repetitions per tuning task and 3 orderings per audit document. Search uses 1 run per question, averaging F1 over 39 whois and 34 whatis questions separately.

Task AVG weights the 3 task-family means equally for tuning, whois and whatis F1 equally for search, and label and exact evidence-set accuracy equally for audit. For each model and regime, we compute sample standard deviation across the 3 repetition or ordering means, first averaging 9 tasks (tuning) or 13 documents (evidence audit) within each repetition or ordering. Across 18 model–regime cells, these standard deviations range from 0.23 to 21.13 for Gap Score, 0.69% to 13.07% for label accuracy, and 0.26% to 13.72% for exact evidence-set accuracy.

C.2 Model Selection and Tied Scores

Model selection uses unrounded task means for the same 6 models. Models tied for the highest Free Feedback mean are eligible, with separate regrets. For NASBench-201's CIFAR-10 task, Kimi K3 and Gemini 3.8 Flash both score 100 under Free Feedback. Their Tight Budget scores are 90.21 and 95.67, versus GPT-5.6-sol's 97.65, for regrets of 7.45 and 1.99, respectively. Section 3.2 reports the Kimi K3 choice. On ParamNet's Adult task, GPT-5.6-sol is the unique Free Feedback leader; its Tight Budget score of 72.01 is below Gemini 3.8 Flash's 87.23, giving a regret of 15.22.

C.3 Feedback Acquisition and Termination

Tables 2 and 3 include all completed trajectories, including those without a scoreable final answer. Returned HPO evaluations count finite, agent-visible performance results; search coverage counts distinct visible articles. Budget termination means reaching the spending boundary, while autonomous submission means an explicit final answer before forced submission. Budget utilization is averaged over autonomous submissions and uses charges for returned evaluations in HPO.

Search repetition rates pool article deliveries per regime. A query is new if its trimmed, lowercased string has not returned an article in that trace. Free Feedback includes 1,160 repeats among 6,567 article deliveries; 1,144 follow new queries, yielding 17.7% and 98.6% in Table 3.

DeepSeek V4 Flash's HPO AVG rises from 61.1 under Free Feedback to 75.8 under Tight Budget as scoreable results increase from 20 to 25 of 27 runs. Among the 19 task–repetition pairs scoreable in both regimes, mean Gap Score instead falls from 85.8 to 78.9. The aggregate improvement is driven by final-result availability, while the paired subset shows weaker configuration performance.

C.4 Answer and Evidence Quality

Table 4 averages hypotheses within each document and ordering, then orderings, documents, and models with equal weight at each level. The nonempty-reference subset also weights documents equally. Missing evidence omits a required identifier; extra evidence adds an incorrect one. Both can occur in one answer. Among nonempty references, extra evidence rises from 3.7% under Free Feedback to 12.5% under Tight Budget, alongside the larger missing-evidence increase in Sec. 4.3.

This diagnostic pairs Free and Tight by model, document, ordering, and hypothesis. Free must receive an incomplete-evidence response and finish with a correct label and exact evidence; Tight must submit the same initial partial set and correct label without visible feedback for that hypothesis. Of the 272 matched presentations across 12 documents, 264 include explicit confirmation of complete evidence under Free. These matched observations associate verification with evidence completion.

C.5 Reasoning Effort under Feedback Constraints

Reasoning effort helps with adequate feedback, but Max cannot recover performance lost to costly feedback. We compare Low and Max effort for GLM-5.3 across all 3 feedback regimes, using 9 tuning tasks with 3 repetitions, 73 search questions, and 13 audit documents with 3 orderings per regime. Both use thinking, temperature 1, top-p 0.95, and the 32,768-token output cap; only the requested effort changes. Search F1 here weights all 73 questions equally, whereas Table 1 weights the whois and whatis family means equally. Figure 6 shows Max ahead in 7 of 9 settings: Search F1 rises from 49.4% to 60.6% under Free Feedback, from 41.3% to 55.7% under Moderate Budget, and from 13.9% to 18.1% under Tight Budget. Under Tight Budget, however, Low exceeds Max in Audit evidence accuracy (58.8% versus 57.9%) and HPO Gap Score (86.1 versus 83.6).

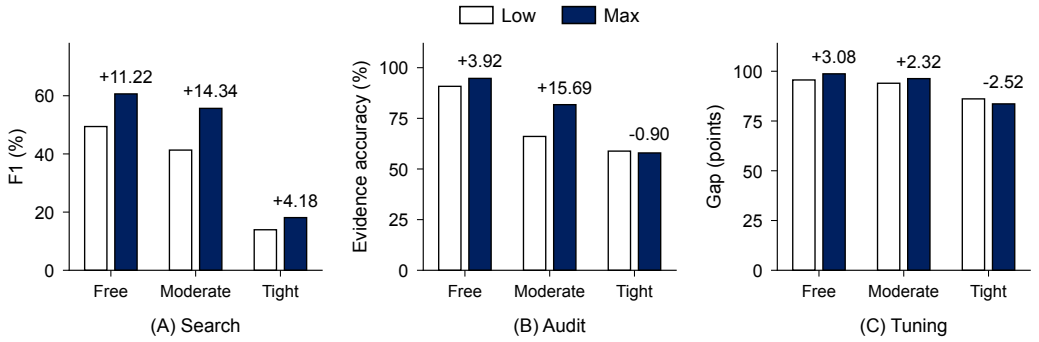


Figure 6: Low versus Max reasoning effort with GLM-5.3 ($N = 1$) under 3 feedback budgets. Bars show mean scores; labels show $\Delta = \text{Max} - \text{Low}$ on each panel's score scale.

C.6 Feedback Cost and Model Response Time

Figure 7 compares assigned feedback costs with measured DeepSeek V4 Flash response latencies under Moderate Budget. Feedback contributes 73.4%–95.5% of the accounted total on ParamNet and more than 99% on all 6 NASBench tasks. For NASBench, eliminating model latency would reduce this total by less than 1%, highlighting the importance of feedback acquisition costs. We used 4 independent replicas, each with 8 NVIDIA H200 GPUs (TP8/PP1/EP1), running SGLang 0.5.17.

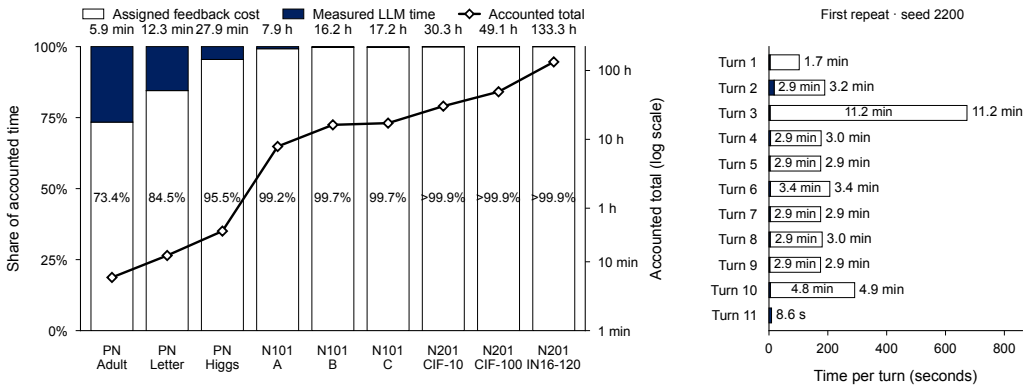


Figure 7: Feedback-cost and response-time breakdown for DeepSeek V4 Flash ($N = 1$, Moderate Budget). Left: task means over 3 repetitions (PN: ParamNet; N101/N201: NASBench-101/201). Right: 11 calls in the first ParamNet Higgs repetition. Accounted time sums assigned feedback costs and client-observed response latencies; it is not measured wall-clock runtime.

D POOLACT IMPLEMENTATION AND ANALYSIS

D.1 EXPERIMENTAL PROTOCOL

Pool size N denotes the number of agents solving one evaluation instance. Table 5 uses $N = 4$: 39 whois questions, 13 audit documents, and 3 repetitions per NASBench-101:A/B/C task for each model, strategy, and budget. Per-agent feedback ceilings and the 30-step limit match across strategies. Shared-graph versions and Gemini serving configurations vary across strategies in the cross-model tuning comparison; Figure 4 holds the implementation fixed. Search and audit report majority-vote F1 and evidence accuracy; tuning averages mean-individual Gap Score over repetitions and tasks.

Scaling and ablations use Max effort with Table 9's settings. Cached and POOLACT evaluate each audit document at $N \in \{2, 4, 6, 8\}$. Naive averages majority-vote scores over all size-N subsets of one $N = 8$ group per document; $N = 1$ uses mean individual scores. The subsets are not independent repetitions. Table 5's GLM-5.3 Tight Budget results and Figure 4 share task instances, repetition counts, and scores, including Naive audit subset averaging. Scaling shares these $N = 4$ audit scores.

D.2 ALGORITHM

```
inputs: N agents, cache C, graph G, lock mu
for each agent in parallel:
  while no final answer and fewer than 30 steps:
    lock, summarize G, and select an action or final answer
    if final answer: unlock and stop this agent
    mark action as in progress, then unlock
    if action is in C:
      return C[action] without charge and record completion in G
    otherwise:
      execute action and apply the budget boundary
      if returned: cache the observation and record completion in G
      if withheld: clear the pending mark, publish no observation,
        request the agent's final answer, and stop this agent
  if no final answer: request one at the step cap
return the aggregate of the final answers
```

D.3 ACQUISITION METRICS

For Table 6, requests match on hypothesis identifier and unordered evidence set. Repetition rates are computed per pool and averaged over documents and models. Visible replies include cache hits, negative judgments, and errors, excluding withheld observations. Coverage counts distinct parseable hypothesis IDs with a reply, regardless of verification success; graph summaries add no replies.

E TRAJECTORY EXAMPLES

These examples show each environment's system prompt, task input, tool calls, feedback, and final submission. They reproduce single-agent runs; bracketed omissions are editorial, and internal reasoning is omitted. Native calls show the function name and JSON arguments.

E.1 HYPERPARAMETER TUNING

NASBench-201, CIFAR-10; DeepSeek V4 Flash; Tight Budget ($\beta = 3$), seed 2200. We show all 4 calls and the final submission. The 6 edges share choices; 5 parameter-range lines are omitted.

System prompt (complete)

```
Interaction protocol: use the API's supplied function tools to acquire feedback. A
    textual Action directive or an imitation of a tool result is not a tool call.
    Make at most one function call per assistant turn. After receiving the real tool
    result, decide whether another observation is useful. When ready, submit your
    final answer in the task's required format (Answer: followed by the answer is
    also accepted). Do not invent observations or claim an unexecuted call succeeded.
```

Task input (excerpt)

```
Goal: tune the system for high performance.
Parameter ranges:
- 1<-0: choices=['none', 'skip_connect', 'nor_conv_1x1', 'nor_conv_3x3',
    'avg_pool_3x3'] (default=none)
[... five further parameter-range lines omitted ...]

[... task note omitted ...]
Call the evaluate_config function with arguments {"param": value, ...} using a native
    tool call.
Answer format: Answer: {"param": value, ...}
```

Generation and feedback

```
Generation 1 (native tool call): evaluate_config
{"1<-0": "none", "2<-0": "none", "2<-1": "none",
 "3<-0": "none", "3<-1": "none", "3<-2": "none"}
Feedback 1:
Observation: perf=0.097120, cost=3641s [time_left=30289s]

Generation 2 (native tool call): evaluate_config
{"1<-0": "nor_conv_3x3", "2<-0": "nor_conv_3x3", "2<-1": "nor_conv_3x3",
 "3<-0": "nor_conv_3x3", "3<-1": "nor_conv_3x3", "3<-2": "nor_conv_3x3"}
Feedback 2:
Observation: perf=0.901987, cost=13525s [time_left=16764s]

Generation 3 (native tool call): evaluate_config
{"1<-0": "nor_conv_3x3", "2<-0": "skip_connect", "2<-1": "nor_conv_3x3",
 "3<-0": "nor_conv_3x3", "3<-1": "skip_connect", "3<-2": "nor_conv_3x3"}
Feedback 3:
Observation: perf=0.898267, cost=9682s [time_left=7082s]

Generation 4 (native tool call): evaluate_config
{"1<-0": "nor_conv_3x3", "2<-0": "nor_conv_3x3", "2<-1": "skip_connect",
 "3<-0": "nor_conv_3x3", "3<-1": "nor_conv_3x3", "3<-2": "nor_conv_3x3"}
Feedback 4:
Observation: [over-budget -- result withheld. This evaluation reached or exceeded the
    time budget.]
```

Budget-stop instruction and final generation

```
System: Loop aborted (Time budget exceeded). Respond immediately with Answer: <your
    final choice> and no other text.

Answer: {"1<-0": "nor_conv_3x3", "2<-0": "nor_conv_3x3", "2<-1": "nor_conv_3x3",
 "3<-0": "nor_conv_3x3", "3<-1": "nor_conv_3x3", "3<-2": "nor_conv_3x3"}
```

E.2 MULTI-HOP SEARCH

Fictional-wiki question 13; DeepSeek V4 Flash; Moderate Budget ($\beta = 10$), seed 2200. We show all 3 retrievals and the final answer. Excerpts retain chain facts; the model received full articles.

System prompt (complete)

```
Interaction protocol: use the API's supplied function tools to acquire feedback. A
    textual Action directive or an imitation of a tool result is not a tool call.
    Make at most one function call per assistant turn. After receiving the real tool
    result, decide whether another observation is useful. When ready, submit your
    final answer in the task's required format (Answer: followed by the answer is
    also accepted). Do not invent observations or claim an unexecuted call succeeded.
```

Task input (excerpt)

```
You are answering a question by searching a wiki of fictional characters.

Tool:
- search: look up information in the wiki.
  Input: {"query": "Person Name"} or {"query": "keyword phrase"}
  If the query exactly matches an article title, that article is returned.
  Otherwise, the most relevant article is returned based on keyword matching.


[... strategy steps 1-3 omitted ...]

4. Each search returns one full article. Re-searching a previously returned article
    is free.
5. Your answer should list ALL matching names, separated by commas.

[... invocation example omitted ...]

Question: Who is the mother-in-law of the father of the person whose date of birth is
    1089-03-31?
```

Generation and feedback

```
Generation 1 (native tool call): search
{"query": "1089-03-31"}
Feedback 1 (article excerpt):
Observation: Article: Milton Hess
[...]
The father of Milton Hess is Alberto Hess.
[...]
The date of birth of Milton Hess is 1089-03-31.
[...]
 | cost=305s [time_left=2695s]

Generation 2 (native tool call): search
{"query": "Alberto Hess"}
Feedback 2 (article excerpt):
Observation: Article: Alberto Hess
[...]
The wife of Alberto Hess is Edythe Hess.
[...]
 | cost=318s [time_left=2377s]

Generation 3 (native tool call): search
{"query": "Edythe Hess"}
Feedback 3 (article excerpt):
Observation: Article: Edythe Hess
[...]
The mother of Edythe Hess is Cordelia Ashford.
[...]
 | cost=291s [time_left=2086s]
```

Final generation

```
Cordelia Ashford
```

E.3 EVIDENCE AUDIT

ContractNLI document 12, hypothesis ordering 2; DeepSeek V4 Flash; Moderate Budget ($\beta = 10$), seed 2202. Calls 3–4 follow hypothesis `nda-12`, omitting other calls and the other 16 final entries.

System prompt (complete)

```
Interaction protocol: use the API's supplied function tools to acquire feedback. A
    textual Action directive or an imitation of a tool result is not a tool call.
    Make at most one function call per assistant turn. After receiving the real tool
    result, decide whether another observation is useful. When ready, submit your
    final answer in the task's required format (Answer: followed by the answer is
    also accepted). Do not invent observations or claim an unexecuted call succeeded.
```

Task input (excerpt)

```
You are auditing a legal document against 17 hypotheses.
Your goal: maximize the correctness of ALL labels and evidence IDs in your final
    answer.
For each hypothesis, determine: Entailment, Contradiction, or NotMentioned.
For Entailment/Contradiction, identify the exact evidence segment IDs.

Tool: human_feedback -- verify your proposed evidence for a hypothesis. Can be called
    multiple times.
  Input: {{"nda_id": "...", "evidence_ids": [segment_ids]}}

Answer: {"nda-11": {"label": "...", "evidence_ids": [...]}, ...}
Your final answer MUST include ALL 17 hypotheses.

Document segments (excerpt):
[...]
[25] 1. For the purposes of this Agreement, Information means information relating to
    one party which is made available by such party to the other party for the
    Business Purpose, including but not limited to technical documentation, analyses,
    studies, knowledge and samples but excluding Information which:
[...]
[29] iv. is independently developed by the Receiving Party;
[...]

Hypotheses (excerpt):
[...]
- nda-12: Permissible development of similar information | Receiving Party may
    independently develop information similar to Confidential Information.
[...]
```

Generation and feedback (consecutive calls)

```
Generation (tool call 3): human_feedback
{"evidence_ids": [29], "nda_id": "nda-12"}
Feedback:
Observation: Evidence Incomplete | missing: Definition scope | cost=283s
    [time_left=2108s]

Generation (tool call 4): human_feedback
{"evidence_ids": [25, 29], "nda_id": "nda-12"}
Feedback:
Observation: Evidence Correct | cost=317s [time_left=1792s]
```

Final generation (entry excerpt)

```
Answer: {
  [... other hypothesis entries omitted ...]
  "nda-12": {"label": "Entailment", "evidence_ids": [25, 29]},
  [... other hypothesis entries omitted ...]
}
```