AGENTFUGUE: AGENT SCALING FOR LONG-HORIZON TASKS THROUGH COLLECTIVE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent progress on long-horizon agentic tasks has been driven largely by scaling up individual agents through stronger models, better tools, and more effective scaffolding. In contrast, much less is understood about scaling out: whether multiple peer agents, all targeting the same task, can become an additional source of capability without relying on explicit role specialization or workflow orchestration. We study this question and propose **AgentFugue**, a collective reasoning framework built around a shared reasoning hub. As peer agents explore the same task in parallel, the hub records concise notes on what each agent has established, attempted, or ruled out, and enables each agent to selectively access what other agents have discovered in a form useful for its current search. This design turns otherwise isolated trajectories into a connected ecology of reusable intermediate reasoning without requiring centralized planning. We instantiate the hub as a plug-in communication layer, trained with supervised fine-tuning and end-to-end reinforcement learning. Across the challenging long-horizon settings we study, AgentFugue improves over strong baselines. Our results suggest that collective reasoning can turn scaling out peer agent systems into a distinct source of capability gains, rather than merely a way of spending more compute. Code and anonymized implementation are available at `https://anonymous.4open.science/r/AgentFugue-EB2C`.

1 INTRODUCTION

Recent progress has shown that LLM-based agents can perform remarkably well on complex long-horizon tasks (Nakano et al., 2021; Qiao et al., 2025; Chen et al., 2026; Ye et al., 2025; Wu et al., 2025a; Zheng et al., 2025; Jin et al., 2025a). A key driver of this progress is sustained scaling up along several dimensions, including stronger foundation models (OpenAI, 2023; Team, 2024; DeepSeek-AI, 2025; Team, 2025), better tool use (Schick et al., 2023; Patil et al., 2024; Qin et al., 2024; Wang et al., 2024b; Li et al., 2025d;e; Jin et al., 2025a), and more effective agent scaffolding (Yao et al., 2023b; Shinn et al., 2023; Yao et al., 2023a; Wang et al., 2024a; Jin et al., 2025b). This scaling-up paradigm has substantially expanded what a single agent can do. At the same time, however, it improves the strength of one trajectory rather than the breadth of exploration, leaving open the question of whether complex tasks may also benefit from scaling beyond a single agent.

Prior work has also shown that multi-agent systems can be effective for complex tasks (Wu et al., 2023; Hong et al., 2024; Li et al., 2023), but the dominant emphasis has been on orchestration: assigning different roles to different agents (Qian et al., 2024; Huang et al., 2023; Chen et al., 2024b), decomposing tasks into separate subtasks (Shen et al., 2023; Wang et al., 2023a; Team, 2026), or designing explicit interaction workflows (Zhuge et al., 2024; Liu et al., 2024; Qian et al., 2025; Zhang et al., 2025). A complementary line of work coordinates multiple agents through deliberation (Du et al., 2024; Liang et al., 2024; Chen et al., 2024a). Such approaches improve capability through structured coordination, with different agents contributing in different ways. What remains less understood is whether gains can also arise in a simpler setting, where multiple agents act as peers on the same task rather than being separated by pre-defined responsibilities.

This peer setting creates a different opportunity for capability growth. When multiple agents explore the same task in parallel (Wang et al., 2023b; Brown et al., 2024; Snell et al., 2025; Li et al., 2024), they may uncover different partial reasoning paths, intermediate evidence, or failed branches. We study whether such parallel exploration can itself become a source of additional capability, rather

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.

than merely additional compute. This is the sense in which we use the term *scaling out*: increasing the number or diversity of peer agents working on the same task so that their trajectories can inform and redirect one another. Realizing this benefit, however, is non-trivial. Without communication, multiple agents largely reduce to isolated searches whose results must be merged after the fact (Wang et al., 2023b; Brown et al., 2024; Li et al., 2024; Lee et al., 2026); with unrestricted communication, useful signals can be overwhelmed by raw trajectory noise (Li et al., 2025b), and the diversity of exploration may quickly collapse. We therefore argue that effective scaling out requires a mechanism for *collective reasoning*, through which peer agents can selectively exchange intermediate progress while continuing to explore the same task from different directions. In this sense, collective reasoning is best understood not as a shared conversation (Du et al., 2024; Liang et al., 2024; Chen et al., 2024a), but as a fugue-like structure of parallel search: in the spirit of a Baroque fugue, multiple trajectories remain distinct while still picking up and developing one another's partial progress (Mann, 1987).

To realize this form of collective reasoning, we propose **AgentFugue**, a framework built around a shared reasoning hub. The hub serves as an external communication layer rather than a centralized planner: when an agent completes a coherent episode of interaction, the hub records a compact note about what that agent established, attempted, or ruled out, and later allows other agents to selectively access the parts of that progress that are useful for their own search. Because the hub is attached outside the core policy, similar in spirit to externalized memory modules studied for single-agent settings (Chhikara et al., 2025; Fang et al., 2025; Tan et al., 2026; Xu et al., 2025; Hu et al., 2026), AgentFugue is adaptive to different reasoning agents while preserving the independence of their local trajectories.

This design lets us study two complementary forms of scaling out. In *homogeneous teams*, multiple agents share the same backbone and configuration, so any gain must come from interaction among parallel trajectories rather than from built-in role differences. In *heterogeneous teams*, agents differ in model or setup, making it possible for distinct reasoning biases to complement one another on the same task (Wang et al., 2025). Across both settings, the central empirical question is whether collective reasoning can improve not only team-level success, but also the quality and efficiency of the individual trajectories that make up the team. In our implementation, the shared reasoning hub is optimized separately from the task agents themselves. We instantiate its write and read functions with a moderate-sized language model, then improve them through supervised fine-tuning followed by end-to-end reinforcement learning so that the hub learns not only to summarize intermediate progress, but also to return guidance that is useful inside the full agent loop.

We evaluate AgentFugue on challenging long-horizon benchmarks spanning information seeking, open-ended problem solving, and multi-step web reasoning. Across the settings we study, we observe gains in both homogeneous and heterogeneous teams, supporting the view that peer-agent communication can provide a robust source of capability beyond stronger individual agents alone.

Our contributions are threefold: (1) we identify peer-agent scaling as a distinct setting for long-horizon reasoning, in which multiple agents work on the same task and capability must arise from cross-trajectory reuse rather than role specialization. (2) we propose AgentFugue, a communication framework based on writing, retrieving, and reading shared reasoning episodes, which turns parallel trajectories into a selectively shared reasoning ecology without centralized planning. (3) we study this framework in both homogeneous and heterogeneous teams, with analyses designed to test when scaling out improves per-agent efficiency, when it yields larger team-level gains, and where the communication mechanism breaks down.

2 METHOD

2.1 PROBLEM SETTING

Target knowledge space. Consider a long-horizon task instance x whose solution requires assembling a body of evidence and reasoning that we call the *target knowledge space* $\mathcal{K}^*(x)$. For hard tasks, $\mathcal{K}^*(x)$ is large and structurally complex: it may span multiple evidence types, reasoning chains, and verification steps. In a single-agent run, the agent explores a trajectory τ and accumulates a discovered subspace $\mathcal{K}(\tau) \subseteq \mathcal{K}^*(x)$. Any single trajectory is unlikely to cover $\mathcal{K}^*(x)$ fully, since each run touches a different, partial, and often sub-optimal fragment, partly by skill and partly

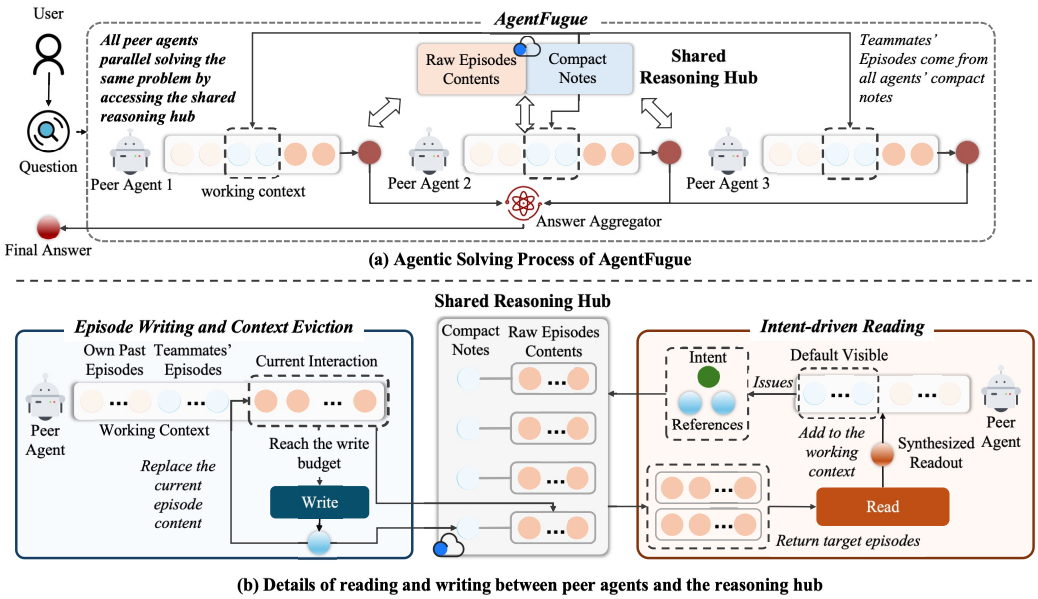


Figure 1: **Overview of AgentFugue.** The top panel illustrates the core idea: peer agents explore the same task in parallel while a shared reasoning hub mediates cross-trajectory communication. The bottom panel details the reasoning hub, including episode writing, context eviction, and intent-driven reading. Once an agent's current interaction reaches the write budget, it is summarized into an episode note and added to the hub; later, another agent can consult relevant teammate episodes and receive synthesized guidance for its ongoing trajectory.

by the luck of which branches happen to be explored. Scaling out, by running multiple peer agents on the same task, creates the opportunity for their discovered fragments to complement one another, but only if the fragments can be shared.

Task, team, and trajectories. We formalize this setting as follows. A team of N agents all target the same task instance x. Agent $i \in \{1, \ldots, N\}$ interacts with the environment through reasoning steps, tool calls, and observations, producing a local trajectory whose prefix up to step t is

$$\tau_{i,t} = \big[(a_{i,1}, o_{i,1}), \ldots, (a_{i,t}, o_{i,t})\big], \tag{1}$$

where $a_{i,t}$ is an action and $o_{i,t}$ the resulting observation. Each trajectory represents a different exploratory path through $\mathcal{K}^*(x)$, with its own discovered subspace $\mathcal{K}(\tau_i)$.

Shared reasoning hub. To connect these scattered fragments, the team is augmented with a *shared reasoning hub* $\mathcal{H}$. As shown in Figure 1, the hub sits alongside the peer agents as a team-level communication interface: it compresses completed portions of each agent's reasoning history into reusable notes and allows agents to consult one another's progress during search. Its role is not to replace local reasoning or to centrally orchestrate the team, but to make intermediate discoveries produced by one trajectory selectively available to others, thereby expanding each agent's effective knowledge space beyond what its own trajectory covers.

Episodes. To make partial progress shareable, we divide each local trajectory into completed *episodes*. An episode $\epsilon_{i,e}$ is a contiguous chunk of interaction history determined by a fixed local context budget:

$$\epsilon_{i,e} = \big[(a_{i,t_s}, o_{i,t_s}), \ldots, (a_{i,t_e}, o_{i,t_e})\big]. \tag{2}$$

Once the active context reaches the budget, the accumulated segment is summarized and written to the hub. Episodes are therefore the units through which an agent's partial progress becomes visible to the rest of the team through $\mathcal{H}$. At any point during search, an agent may either continue along its own local trajectory or consult $\mathcal{H}$ to access relevant progress produced by other team members.

This formulation subsumes several useful limiting cases. When $N{=}1$, it reduces to a single reasoning agent coupled with an external memory-like module. When $N{>}1$ but no agent consults the hub, it reduces to multiple isolated trajectories that share compute but not information. Our main interest lies between these extremes: peer-agent teams in which multiple agents pursue the same task while selectively reusing one another's intermediate reasoning.

Two forms of scaling out. We study this setting in two forms. In *homogeneous* teams, all agents share the same model and configuration, so any gains must arise from cross-trajectory interaction rather than built-in agent differences. In *heterogeneous* teams, agents differ in model backbone or prompting configuration, introducing systematic diversity beyond stochastic variation: different models carry different reasoning biases, knowledge distributions, and failure modes, so the hub can additionally mediate complementary strengths across the team.

From isolated fragments to connected knowledge. Without communication, the team's collective knowledge $\bigcup_i \mathcal{K}(\tau_i)$ exists only in aggregate: no individual agent can access another's discoveries, so each remains limited to its own fragment. The role of $\mathcal{H}$ is to connect these scattered fragments by making useful portions of one trajectory selectively available to another, expanding each agent's effective knowledge space beyond $\mathcal{K}(\tau_{i,t})$ alone. This perspective clarifies both the promise and the limit of scaling out: adding agents increases the diversity of discovered fragments, but the marginal gain depends on whether new trajectories reach genuinely new regions of the task-relevant knowledge needed to solve x, denoted conceptually by $\mathcal{K}^*(x)$, and whether the hub can surface those regions when they are needed. The rest of this section describes the hub mechanism that operationalizes this view (§2.2) and how we optimize it (§2.3).

2.2 Shared Reasoning Hub

AgentFugue operationalizes the shared reasoning hub through two operations: *episode writing*, which compresses completed trajectory segments into reusable notes, and *intent-driven reading*, which lets agents inspect and synthesize relevant teammate episodes on demand.

Episode writing and context eviction. As illustrated in the top-right panel of Figure 1, agent i's local context window accumulates reasoning steps, tool calls, and observations until it reaches a fixed write budget, at which point the current segment is closed as an episode $\epsilon_{i,e}$. The hub model then compresses the episode into an *episode note*:

$$z_{i,e} = M_{\text{write}}(\epsilon_{i,e}), \tag{3}$$

which captures the team-relevant content of that episode: what was established, what evidence was collected, what was attempted, and which branches were ruled out. Once the note is written, the raw episode content in the agent's working context is *evicted and replaced* by its episode note $z_{i,e}$. This serves a dual purpose: it compresses the agent's own history to free context capacity for continued exploration, and it produces a representation suitable for sharing with other agents. The full episode content $\epsilon_{i,e}$ is retained in the hub's storage for later deep reading.

At any point during search, agent i's working context therefore takes the form:

$$\mathcal{C}_{i,t} = \big[\ \underbrace{z_{i,1}, \ldots, z_{i,e-1}}_{\text{own episode notes}},\ \underbrace{z_{j_1,e'}, \ldots, z_{j_k,e''}}_{\text{teammates' notes}},\ \underbrace{\tau_{i,t}^{\text{active}}}_{\text{current interaction}}\ \big], \tag{4}$$

where the first group contains episode notes summarizing agent i's own completed episodes, the second group contains episode notes from other agents that have been made visible through prior hub interactions, and $\tau_{i,t}^{\text{active}}$ is the current unfinished interaction segment. This design keeps the working context bounded even as total reasoning effort grows, while exposing a structured view of the team's collective progress.

Intent-driven reading. As shown in the bottom-right panel of Figure 1, agents do not passively receive all teammate episode notes. Instead, when agent i judges, based on its current context $\mathcal{C}_{i,t}$, that consulting a teammate's work in greater depth would be useful, it issues a structured request to the hub with two components: an *intent* $q_{i,t}$ describing what kind of information is needed, and a set of *episode references* $\mathcal{E}_{i,t} \subseteq \{\epsilon_{j,e} \mid j \neq i\}$ indicating which teammates' episodes it wants to

inspect in full. The agent selects these references based on the episode notes already visible in $\mathcal{C}_{i,t}$: for example, an episode note may indicate that another agent found evidence related to the current search direction, prompting a request for the original episode.

Given this request, the hub retrieves the full raw content of the referenced episodes from its storage and synthesizes them in light of the intent:

$$r_{i,t} = M_{\text{read}}\big(q_{i,t},\ \ \{\epsilon_{j,e} : \epsilon_{j,e} \in \mathcal{E}_{i,t}\}\big). \tag{5}$$

The resulting readout $r_{i,t}$ is a focused piece of evidence or guidance tailored to the requesting agent's current need, which is appended to $\mathcal{C}_{i,t}$. In this design, episode notes provide coarse awareness and help the agent identify which episodes are worth inspecting, while the hub performs the actual synthesis over the raw referenced content. This two-level design, with episode notes for broad awareness and intent-driven reading for selective depth, avoids both extremes of no communication and full broadcast. Agents maintain a lightweight overview of team progress through episode notes and can drill into specific episodes when deeper information is needed.

Distinction from nearby paradigms. The write/read mechanism differs from several adjacent settings in important ways. Unlike single-agent memory, notes are written to support cross-agent reuse, not just the originating trajectory. Unlike multi-agent debate or group chat, agents are not forced into synchronized turn-taking or a shared conversational context. Unlike best-of-N sampling, trajectories influence one another *before* completion through reusable intermediate progress. And unlike RAG-style retrieval over a static corpus, the read path is intent-driven and synthesizes *raw episode content* on demand rather than returning pre-formed passages.

2.3 Hub Optimization

The hub is initialized from a Qwen3.5-9B backbone, with separate M_{write}/M_{read} instances from the same checkpoint, and optimized in two stages. We keep the task agents frozen throughout hub training: the hub is the only learned component in both stages.

Training data and teacher traces. We collect training trajectories from OpenSeeker (Du et al., 2026) and OpenResearcher (Li et al., 2026). Because the two sources differ in trajectory length and final-answer reliability, we remove trajectories that are too short to exhibit substantive long-horizon interaction and trajectories whose final answers do not match the verified gold answers. This filtering yields $2{,}000$ high-quality training trajectories. GPT-5.4 is used as the teacher model to generate a reference note for each completed episode and a reference readout conditioned on the current intent and the referenced raw episodes. These paired write/read traces supervise both hub operations.

Supervised fine-tuning. GPT-5.4 produces reference notes z^* for each completed episode and reference readouts r^* for each read request, yielding $\mathcal{D}_{\text{write}}$ and $\mathcal{D}_{\text{read}}$. Both heads are trained jointly with the standard LM loss:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{\mathcal{D}_{\text{write}}}\big[\log \pi_\theta(z^* \,|\, \epsilon)\big] - \mathbb{E}_{\mathcal{D}_{\text{read}}}\big[\log \pi_\theta(r^* \,|\, q, \mathcal{E})\big]. \tag{6}$$

Group relative policy optimization. We then align the hub with downstream task success via GRPO in the full multi-agent loop, keeping task agents frozen. For instance x, sample G candidate hub outputs $\{y_g\}$, run the loop with each, observe rewards $\{R_g\}$, form group-relative advantages $\hat{A}_g = (R_g - \text{mean})/(\text{std} + \varepsilon)$, and optimize:

$$\mathcal{J}_{\text{GRPO}} = \mathbb{E}\Big[\tfrac{1}{G}\sum_g \min\big(\rho_g \hat{A}_g,\ \text{clip}(\rho_g, 1{-}\delta, 1{+}\delta)\hat{A}_g\big)\Big] - \beta\, D_{\text{KL}}\big[\pi_\theta \,\|\, \pi_{\text{ref}}\big], \tag{7}$$

with $\rho_g = \pi_\theta(y_g)/\pi_{\text{old}}(y_g)$, π_{ref} the SFT checkpoint, and R_g combining task success with a brevity bonus that favors hub outputs leading to shorter effective search paths. Because task agents are frozen, GRPO pressure lands on the communication layer itself.

3 EXPERIMENTS

3.1 DATASETS

We evaluate on three benchmarks chosen to stress complementary aspects of long-horizon agentic reasoning: **BrowseComp** (Wei et al., 2025), which requires deep multi-hop web search and cross-document evidence aggregation toward a short factual answer; **WideSearch** (Wong et al., 2025), which rewards *breadth* of evidence collection rather than depth, asking agents to enumerate and consolidate many parallel pieces of information; and **HLE (Humanity's Last Exam)** (Phan et al., 2025), an expert-authored multi-domain reasoning benchmark whose questions stress deliberate multi-step reasoning rather than web navigation. For all three benchmarks we follow the official judging protocol. For evaluation efficiency and to keep the compute budget manageable, on BrowseComp and HLE we follow prior work (Li et al., 2025e; Lee et al., 2026; Feng et al., 2026) and evaluate on a 200-question random sample rather than the full test set; WideSearch is used in full. More details are deferred to Appendix B.

3.2 BASELINES

We compare against three groups of systems (Appendix C); all multi-agent systems share the same per-agent tool stack and interaction budget so that any difference reflects coordination, not capability.

Single-agent ReAct. Frontier models in a standard ReAct (Yao et al., 2023b) loop with the same tool stack as AgentFugue, isolating how far "scaling up" a single agent goes: **Claude-Opus-4.5**, **Kimi-K2.5** (Team, 2026), **Qwen3.5-35B-A3B**, **GLM-4.7**, and **DeepSeek-v4-Flash**.

Single-agent DeepResearch. Single-agent systems with extended scaffolding (search planning, summary memory, iterative refinement) for long-horizon web research: **WebThinker** (Li et al., 2025e), **WebSailor** (Li et al., 2025c), **AgentFold** (Ye et al., 2025), **IterResearch** (Chen et al., 2026), **Tongyi-DeepResearch** (Li et al., 2025a), and **OpenAI DeepResearch** (OpenAI, 2025).

Multi-agent systems. Direct alternatives that also run multiple peer agents per task: **Naive-Multi-Agent**, a plan/parallel-search/aggregate pipeline through a meta-agent, and **Swarm-Multi-Agent**, the swarm setting from Kimi-K2.5 (Team, 2026) with `create_subagent`/`assign_task` tools. Against both, **AgentFugue** replaces the central meta-agent with a shared reasoning hub: communication is *horizontal* between peers rather than *vertical* through a planner, and agents exchange intermediate progress *during* exploration rather than only at aggregation. The hub is initialized from Qwen3.5-9B and trained as in §2.3 (Appendix A). Throughout Table 1 a team-level prediction is the answer of the agent with the highest self-reported confidence; alternative aggregators are studied in §3.4 (Appendix D).

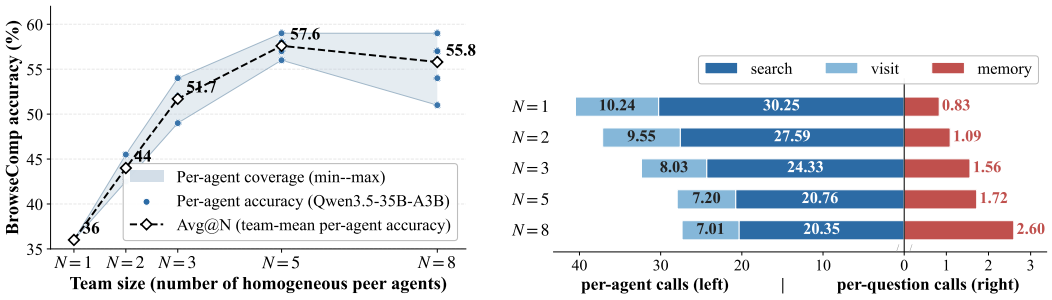
3.3 MAIN RESULTS

Table 1 reports BrowseComp, WideSearch, and HLE accuracy for all systems. AgentFugue delivers the strongest numbers on every benchmark, and we highlight two takeaways.

- **Consistent dominance over every multi-agent baseline under both backbones.** Under Qwen3.5-35B-A3B, AgentFugue reaches 54.4 Avg, $+5.4$/$+5.9$ over Swarm-/Naive-Multi-Agent; under DeepSeek-v4-Flash it reaches 65.0 Avg, $+7.4$/$+7.7$ over the corresponding baselines. The lead holds on every benchmark, showing the gain comes from the shared-hub coordination itself rather than any single benchmark's idiosyncrasy.
- **Gains generalize across heterogeneous benchmarks.** The improvement is not confined to one task type. Compared with the same-backbone Swarm baseline, AgentFugue/DeepSeek improves BrowseComp by $+15.0$ ($56.2 \rightarrow 71.2$, retrieval-heavy), HLE by $+5.5$ ($44.0 \rightarrow 49.5$, reasoning-centric), and remains ahead on the already-saturated WideSearch ($72.7 \rightarrow 74.2$, breadth-oriented); the Qwen-backed team shows the same monotone pattern across all three benchmarks. Stable improvements across retrieval, reasoning, and breadth benchmarks indicate that the shared reasoning hub is a generic coordination primitive rather than a benchmark-specific trick.

Table 1: Main results on BrowseComp, WideSearch, and HLE. Results marked with † are cited from original papers. **Bold** marks the best score within each backbone group of the multi-agent block.

System	Backbone	Benchmark			Avg
		BrowseComp	WideSearch	HLE	
Single-Agent: LLM ReAct					
Claude-Opus-4.5†	–	37.0	–	43.4	–
Qwen3.5-35B-A3B	–	36.0	65.6	34.0	45.2
GLM-4.7	–	43.5	65.4	37.2	48.7
DeepSeek-v4-Flash	–	56.2	69.7	42.0	56.0
Kimi-K2.5† (Team, 2026)	–	60.6	72.7	50.2	–
Single-Agent: DeepResearch					
WebThinker† (Li et al., 2025e)	QwQ-32B	2.8	–	15.8	–
WebSailor† (Li et al., 2025c)	QwQ-32B	10.5	–	9.6	–
AgentFold† (Ye et al., 2025)	Qwen3-30B-A3B	36.2	62.1	–	–
IterResearch† (Chen et al., 2026)	Qwen3-30B-A3B	37.3	–	28.8	–
Tongyi-DeepResearch† (Li et al., 2025a)	Qwen3-30B-A3B	43.4	–	32.9	–
OpenAI DeepResearch† (OpenAI, 2025)	–	51.5	–	26.6	–
Multi-Agent (team size N=2)					
Naive-Multi-Agent	Qwen3.5-35B-A3B	44.5	68.9	32.2	48.5
	DeepSeek-v4-Flash	59.0	69.3	43.5	57.3
Swarm-Multi-Agent	Qwen3.5-35B-A3B	45.2	70.4	31.5	49.0
	DeepSeek-v4-Flash	56.2	72.7	44.0	57.6
AgentFugue (Ours)	Qwen3.5-35B-A3B	**50.0**	**72.3**	**41.0**	**54.4**
	DeepSeek-v4-Flash	**71.2**	**74.2**	**49.5**	**65.0**



(a) Per-agent accuracy with team size

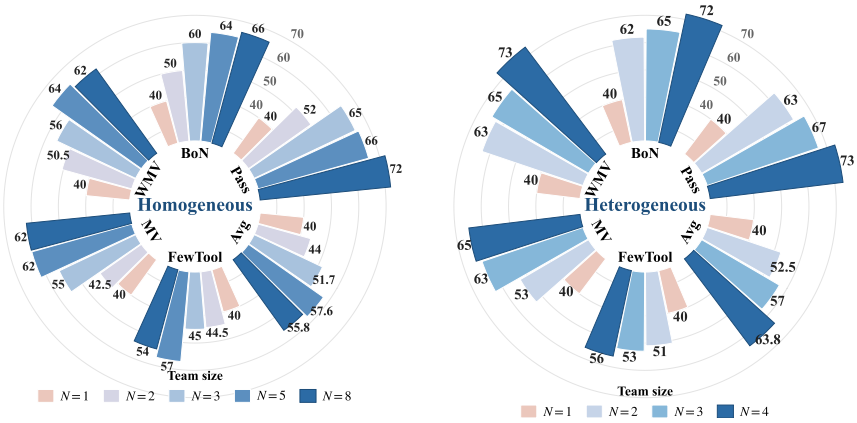
(b) Workload vs. coordination cost

Figure 2: **Homogeneous scaling on BrowseComp (Qwen3.5-35B-A3B, $N \in \{1, 2, 3, 5, 8\}$).** (a) Per-agent accuracy and team-mean Avg@N as the team grows. (b) Per-agent search/visit calls (cool) vs. per-question memory calls (warm); larger teams shift effort from isolated exploration to shared coordination.

3.4 Scaling Behavior: Homogeneous Teams

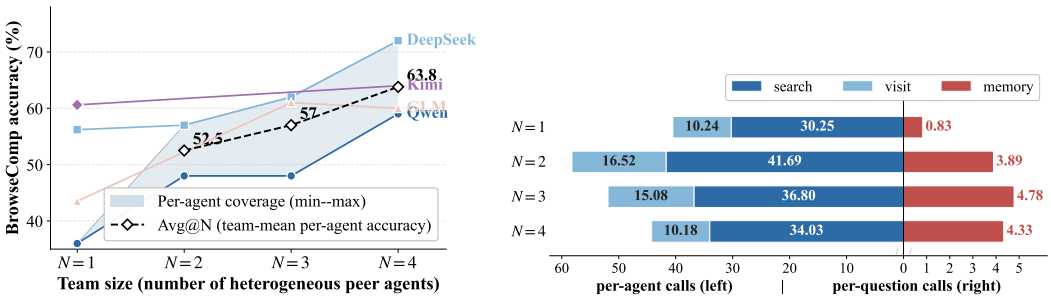
Having fixed N=2 for the head-to-head comparison above, we now ask whether *adding more copies of the same agent*—connected through the shared hub—is itself a meaningful scaling axis. To remove cross-model diversity as a confounder, all peers use the same Qwen3.5-35B-A3B backbone and we vary only $N \in \{1, 2, 3, 5, 8\}$ on the BrowseComp 100-question subset.

Per-agent quality benefits first, then saturates around N=5 (Fig. 2a). The dashed Avg@N curve climbs sharply at small N and plateaus by N=5: each peer absorbs about as much useful context from the hub as it can hold. Yet the per-agent coverage band (min–max across peers) stays wide even at the largest team size, so trajectories remain *diverse* rather than collapsing onto the average—exactly the regime where aggregation-side scaling continues to pay off.



(a) Homogeneous teams ($N \in \{1, 2, 3, 5, 8\}$) (b) Heterogeneous teams ($N \in \{1, 2, 3, 4\}$)

Figure 3: **Aggregator metrics across team sizes** (Pass, BoN, MV, WMV, FewTool, Avg). Each spoke is one heuristic rule; bars within a spoke sweep N from light to dark.



(a) Per-model trajectories with team size (b) Workload vs. coordination cost

Figure 4: **Heterogeneous scaling on BrowseComp (Qwen → +DeepSeek-v4-Flash → +GLM-4.7 → +Kimi-K2.5).** (a) Per-model per-agent accuracy as each backbone joins; markers at N=1 are standalone baselines. (b) Per-agent search/visit vs. per-question memory calls.

The hub trades isolated exploration for shared coordination (Fig. 2b). Per-agent search and visit calls drop monotonically from N=1 to N=8: each peer is *cheaper* because it inherits partial work through the hub. Per-question memory traffic moves the opposite way, growing several-fold over the same range. The cool exploration bars contract while the warm memory bar extends, exposing a clean shift from many isolated explorations toward a smaller amount of structured coordination.

Scaling holds under every aggregation rule (Fig. 3a). Stepping back from individual trajectories to the team's aggregated output, we restrict ourselves to model-free heuristic aggregators (continuing ğ3): Pass@N (oracle coverage), BoN@N (the deployed confidence-highest selector), MV/WMV@N, and a FewTool@N fallback (Appendix D). Each spoke of Fig. 3a is one rule; bars sweep N from light to dark. Every aggregator improves substantially with N, so scaling does not depend on the choice of selector. The persistent Pass–BoN gap further shows that the right answer is in the rollouts more often than the deployed selector picks it, so smarter aggregation is a free axis of headroom on top of the same hub.

3.5 Heterogeneous Teams: Stronger Models Pull Up the Group

We next ask whether the same hub generalizes to teams of *different* backbones. Starting from Qwen3.5-35B-A3B, we successively add DeepSeek-v4-Flash, GLM-4.7, and Kimi-K2.5 (N=1→4), so each team is a strict superset of the previous and any change comes from *introducing* a qualitatively different peer.

Every model individually benefits, with weaker peers gaining the most (Fig. 4a). At the largest team size, every model beats its standalone $N{=}1$ score, with gains roughly inversely related to its baseline. Weaker peers receive the largest, often double-digit, improvements, while even Kimi-K2.5 continues to improve after joining the hub. Small non-monotonicities reflect temporary uncertainty in the shared notes.

Heterogeneity injects fresh exploration, then triggers far heavier hub usage than homogeneous teams (Fig. 4b). Search+visit rises when the first different peer joins, as the new backbone follows different search directions, then contracts as agents inherit one another's findings. Complementary backbones also produce several times more memory traffic than homogeneous teams, indicating that their notes are more novel and useful.

Aggregator-side scaling persists, with a tighter coverage/consensus gap (Fig. 3b). All five training-free aggregators improve monotonically with N, so the gain comes from the rollouts rather than a particular selector. Heterogeneous teams rise faster and show a tighter coverage/consensus gap as peers agree on shared evidence, supporting the claim that the hub makes agent count a genuine scaling axis.

3.6 Ablation Studies

We sweep the per-agent hub budget across $\{16, 32, 64, 96, 128\}$K at $N{=}2$ on BrowseComp (Appendix H, Table 2). Every aggregator peaks at 32K and degrades at both extremes: small budgets truncate evidence, whereas large budgets dilute attention with stale content. This consistent trend suggests that the optimum is a property of the communication channel, so the 64K trigger used in the main tables is conservative; case studies and broader failure modes appear in Appendices E and G.

4 Related Work

Multi-agent LLM collaboration. Most prior multi-agent systems specialize roles, orchestrate workflows, or use debate and learned topologies (Wu et al., 2023; Hong et al., 2024; Li et al., 2023; Chen et al., 2024b; Hu et al., 2025; Qian et al., 2024; Huang et al., 2023; Du et al., 2024; Liang et al., 2024; Chen et al., 2024a; Zhuge et al., 2024; Liu et al., 2024; Qian et al., 2025). Their intermediate discoveries often remain local; AgentFugue instead uses a shared hub that records and selectively serves evidence conditioned on the reader's intent.

Test-time scaling. Test-time scaling increases either trajectory depth through reasoning, search, or folding (Wei et al., 2022; Kojima et al., 2022; Yao et al., 2023a; Snell et al., 2025; Muennighoff et al., 2025; DeepSeek-AI, 2025; Wu et al., 2025b; Sun et al., 2025; Ye et al., 2025; Qian et al., 2026; Chen et al., 2026), or breadth through repeated sampling and aggregation (Wang et al., 2023b; Brown et al., 2024; Qi et al., 2025; Zhao et al., 2025; Lee et al., 2026; Qiao et al., 2025; Li et al., 2025b; Chang et al., 2026). These rollouts are usually opaque during exploration; AgentFugue connects them through a trained communication hub.

5 Conclusion

We study scaling out for long-horizon reasoning: multiple peer trajectories improve one another through a shared hub that writes compact episode notes and supports intent-driven reading. Gains across homogeneous and heterogeneous teams show capability beyond independent sampling, especially when complementary agents exchange new evidence. The effect is not merely a consequence of spending more tokens: per-agent exploration falls as the hub reuses prior work, while useful memory traffic grows with team diversity. Context budgeting and aggregation remain clear headroom: more memory is not always better, and selectivity and reliability remain open challenges. In particular, improving intent-conditioned retrieval and making the aggregator uncertainty-aware should let future systems capture more of the correct evidence already present in the shared rollouts. These directions could make scaling out a practical complement to scaling up for long-horizon research tasks.

AI USE STATEMENT

Generative AI models are used as experimental components of this work: the peer agents, shared reasoning hub, baseline systems, and some benchmark judges are language models, and GPT-5.4 is used to generate reference notes and intent-conditioned readouts for hub training. These uses are part of the method and evaluation and are described in the main text and Appendix A. During manuscript preparation, generative AI was used only to polish author-written text (proofreading, grammar correction, and minor rewording). It was not used to generate the research idea, formulate the hypotheses, design the experiments, derive or interpret the reported results, create figures or tables, or author technical claims. The authors reviewed all AI-assisted text and take responsibility for the final manuscript and all artifacts.

ETHICS STATEMENT

This work uses public benchmark tasks and does not involve human participants, private personal data, or the collection of sensitive user information. The proposed system can nevertheless make multi-agent evidence gathering and coordination more capable, which could be misused for surveillance, manipulation, or generation of deceptive content. The shared hub can also propagate biased or incorrect intermediate conclusions across agents. We discuss these limitations and potential harms in Appendix G; any deployment should use access controls, monitoring, confidence-aware outputs, and safeguards appropriate to the application.

REPRODUCIBILITY STATEMENT

The method, training-data curation procedure, model initialization, and evaluation protocol are specified in Sections 2.2–2.3 and Appendix A. Appendix B, Appendix C, and Appendix D document the benchmark splits, judging protocols, comparison systems, and aggregation rules, while Appendix F provides the core hub prompts. An anonymized implementation and configuration files can be supplied as supplementary material for review and will be released upon acceptance.

REFERENCES

Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787, 2024. doi: 10.48550/ARXIV.2407.21787. URL https://doi.org/10.48550/arXiv.2407.21787.

Jonathan D. Chang, Andrew Drozdov, Shubham Toshniwal, Owen Oertell, Alexander Trott, Jacob P. Portes, Abhay Gupta, Pallavi Koppol, Ashutosh Baheti, Sean Kulinski, Ivan Zhou, Irene Dea, Krista Opsahl-Ong, Simon Favreau-Lessard, Sean Owen, Jose Javier Gonzalez Ortiz, Arnav Singhvi, Xabi Andrade, Cindy Wang, Kartik Sreenivasan, Sam Havens, Jialu Liu, Peyton DeNiro, Wen Sun, Michael Bendersky, and Jonathan Frankle. KARL: knowledge agents via reinforcement learning. *CoRR*, abs/2603.05218, 2026. doi: 10.48550/ARXIV.2603.05218. URL https://doi.org/10.48550/arXiv.2603.05218.

Guoxin Chen, Zile Qiao, Xuanzhong Chen, Donglei Yu, Haotian Xu, Wayne Xin Zhao, Ruihua Song, Wenbiao Yin, Huifeng Yin, Liwen Zhang, Kuan Li, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Iterresearch: Rethinking long-horizon agents with interaction scaling, 2026. URL https://arxiv.org/abs/2511.07327.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 7066–7085. Association for Computational Linguistics, 2024a. doi: 10.18653/V1/2024.ACL-LONG.381. URL https://doi.org/10.18653/v1/2024.acl-long.381.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent

behaviors. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=EHg5GDnyq1.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In Inês Lynce, Nello Murano, Mauro Vallati, Serena Villata, Federico Chesani, Michela Milano, Andrea Omicini, and Mehdi Dastani (eds.), *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, Frontiers in Artificial Intelligence and Applications, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 11733–11763. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Yuwen Du, Rui Ye, Shuo Tang, Xinyu Zhu, Yijun Lu, Yuzhu Cai, and Siheng Chen. Openseeker: Democratizing frontier search agents by fully open-sourcing training data. *CoRR*, abs/2603.15594, 2026. doi: 10.48550/ARXIV.2603.15594. URL https://doi.org/10.48550/arXiv.2603.15594.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. *CoRR*, abs/2510.18866, 2025. doi: 10.48550/ARXIV.2510.18866. URL https://doi.org/10.48550/arXiv.2510.18866.

Zhaopeng Feng, Liangcai Su, Zhen Zhang, Xinyu Wang, Xiaotian Zhang, Xiaobin Wang, Runnan Fang, Qi Zhang, Baixuan Li, Shihao Cai, Rui Ye, Hui Chen, Yong Jiang, Joey Tianyi Zhou, Chenxiong Qian, Pengjun Xie, Bryan Hooi, Zuozhu Liu, and Jingren Zhou. Agentswing: Adaptive parallel context management routing for long-horizon web agents. *CoRR*, abs/2603.27490, 2026. doi: 10.48550/ARXIV.2603.27490. URL https://doi.org/10.48550/arXiv.2603.27490.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, Zhenrong Cheng, Xuanbo Fan, Jiaxin Guo, Xinlei Yu, Zhenhong Zhou, Zewen Hu, Jiahao Huo, Junhao Wang, Yuwei Niu, Yu Wang, Zhenfei Yin, Xiaobin Hu, Yue Liao, Qiankun Li, Kun Wang, Wangchunshu Zhou, Yixin Liu, Dawei Cheng, Qi Zhang, Tao Gui, Shirui Pan, Yan Zhang, Philip Torr, Zhicheng Dou, Ji-Rong Wen, Xuanjing Huang, Yu-Gang Jiang, and Shuicheng Yan. Memory in the age of AI agents. *CoRR*, abs/2512.13564, 2025. doi: 10.48550/ARXIV.2512.13564. URL https://doi.org/10.48550/arXiv.2512.13564.

Yuyang Hu, Jiongnan Liu, Jiejun Tan, Yutao Zhu, and Zhicheng Dou. Memory matters more: Event-centric memory as a logic map for agent searching and reasoning. *CoRR*, abs/2601.04726, 2026. doi: 10.48550/ARXIV.2601.04726. URL https://doi.org/10.48550/arXiv.2601.04726.

Dong Huang, Qingwen Bu, Jie M. Zhang, Michael Luck, and Heming Cui. Agentcoder: Multi-agent-based code generation with iterative testing and optimisation. *CoRR*, abs/2312.13010, 2023. doi: 10.48550/ARXIV.2312.13010. URL https://doi.org/10.48550/arXiv.2312.13010.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516, 2025a. doi: 10.48550/ARXIV.2503.09516. URL https://doi.org/10.48550/arXiv.2503.09516.

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (eds.), *Companion Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025 - 2 May 2025*, pp. 737–740. ACM, 2025b. doi: 10.1145/3701716.3715313. URL https://doi.org/10.1145/3701716.3715313.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Yoonsang Lee, Howard Yen, Xi Ye, and Danqi Chen. Agentic aggregation for parallel scaling of long-horizon agentic tasks, 2026. URL https://arxiv.org/abs/2604.11753.

Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, Kuan Li, Liangcai Su, Litu Ou, Liwen Zhang, Pengjun Xie, Rui Ye, Wenbiao Yin, Xinmiao Yu, Xinyu Wang, Xixi Wu, Xuanzhong Chen, Yida Zhao, Zhen Zhang, Zhengwei Tao, Zhongwang Zhang, Zile Qiao, Chenxi Wang, Donglei Yu, Gang Fu, Haiyang Shen, Jiayin Yang, Jun Lin, Junkai Zhang, Kui Zeng, Li Yang, Hailong Yin, Maojia Song, Ming Yan, Peng Xia, Qian Xiao, Rui Min, Ruixue Ding, Runnan Fang, Shaowei Chen, Shen Huang, Shihang Wang, Shihao Cai, Weizhou Shen, Xiaobin Wang, Xin Guan, Xinyu Geng, Yingcheng Shi, Yuning Wu, Zhuo Chen, Zijian Li, and Yong Jiang. Tongyi deepresearch technical report. *CoRR*, abs/2510.24701, 2025a. doi: 10.48550/ARXIV.2510.24701. URL https://doi.org/10.48550/arXiv.2510.24701.

Baixuan Li, Dingchu Zhang, Jialong Wu, Wenbiao Yin, Zhengwei Tao, Yida Zhao, Liwen Zhang, Haiyang Shen, Runnan Fang, Pengjun Xie, Jingren Zhou, and Yong Jiang. Parallelmuse: Agentic parallel thinking for deep information seeking. *CoRR*, abs/2510.24698, 2025b. doi: 10.48550/ARXIV.2510.24698. URL https://doi.org/10.48550/arXiv.2510.24698.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: communicative agents for "mind" exploration of large language model society. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=bgzUSZ8aeg.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating superhuman reasoning for web agent. *CoRR*, abs/2507.02592, 2025c. doi: 10.48550/ARXIV.2507.02592. URL https://doi.org/10.48550/arXiv.2507.02592.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 5420–5438. Association for Computational Linguistics, 2025d. doi: 10.18653/V1/2025.EMNLP-MAIN.276. URL https://doi.org/10.18653/v1/2025.emnlp-main.276.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *CoRR*, abs/2504.21776, 2025e. doi: 10.48550/ARXIV.2504.21776. URL https://doi.org/10.48550/arXiv.2504.21776.

Zhuofeng Li, Dongfu Jiang, Xueguang Ma, Haoxiang Zhang, Ping Nie, Yuyu Zhang, Kai Zou, Jianwen Xie, Yu Zhang, and Wenhu Chen. Openresearcher: A fully open pipeline for long-horizon deep research trajectory synthesis. *CoRR*, abs/2603.20278, 2026. doi: 10.48550/ARXIV.2603.20278. URL https://doi.org/10.48550/arXiv.2603.20278.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 17889–17904. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.992. URL https://doi.org/10.18653/v1/2024.emnlp-main.992.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration, 2024. URL https://arxiv.org/abs/2310.02170.

Alfred Mann. *The Study of Fugue*. Courier Corporation, 1987.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 20275–20321. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.1025. URL https://doi.org/10.18653/v1/2025.emnlp-main.1025.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021. URL https://arxiv.org/abs/2112.09332.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

OpenAI. Introducing deep research. https://openai.com/index/introducing-deep-research/, February 2025. Accessed: 2026-05-06.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/e4c61f578ff07830f5c37378dd3ecb0d-Abstract-Conference.html.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra,

Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Summer Yue, Alexandr Wang, and Dan Hendrycks. Humanity's last exam. *CoRR*, abs/2501.14249, 2025. doi: 10.48550/ARXIV.2501.14249. URL https://doi.org/10.48550/arXiv.2501.14249.

Jianing Qi, Xi Ye, Hao Tang, Zhigang Zhu, and Eunsol Choi. Learning to reason across parallel samples for LLM reasoning. *CoRR*, abs/2506.09014, 2025. doi: 10.48550/ARXIV.2506.09014. URL https://doi.org/10.48550/arXiv.2506.09014.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chatdev: Communicative agents for software development. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.810. URL https://doi.org/10.18653/v1/2024.acl-long.810.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large language model-based multi-agent collaboration. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=K3n5jPkrU6.

Hongjin Qian, Zhao Cao, and Zheng Liu. Memobrain: Executive memory as an agentic brain for reasoning. *CoRR*, abs/2601.08079, 2026. doi: 10.48550/ARXIV.2601.08079. URL https://doi.org/10.48550/arXiv.2601.08079.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, Rui Min, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *CoRR*, abs/2509.13309, 2025. doi: 10.48550/ARXIV.2509.13309. URL https://doi.org/10.48550/arXiv.2509.13309.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

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 Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving AI tasks with chatgpt and its friends in hugging face. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/77c33e6a367922d003ff102ffb92b658-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16,*

2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon LLM agent via context-folding. *CoRR*, abs/2510.11967, 2025. doi: 10.48550/ARXIV.2510.11967. URL https://doi.org/10.48550/arXiv.2510.11967.

Jiejun Tan, Zhicheng Dou, Liancheng Zhang, Yuyang Hu, Yiruo Cheng, and Ji-Rong Wen. Memsifter: Offloading llm memory retrieval via outcome-driven proxy reasoning, 2026. URL https://arxiv.org/abs/2603.03379.

Kimi Team. Kimi K2.5: visual agentic intelligence. *CoRR*, abs/2602.02276, 2026. doi: 10.48550/ARXIV.2602.02276. URL https://doi.org/10.48550/arXiv.2602.02276.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

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. *Trans. Mach. Learn. Res.*, 2024, 2024a. URL https://openreview.net/forum?id=ehfRiF0R3a.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=h0ZfDIrj7T.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 2609–2634. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.ACL-LONG.147. URL https://doi.org/10.18653/v1/2023.acl-long.147.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 50208–50232. PMLR / OpenReview.net, 2024b. URL https://proceedings.mlr.press/v235/wang24h.html.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. 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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *CoRR*, abs/2504.12516, 2025. doi: 10.48550/ARXIV.2504.12516. URL https://doi.org/10.48550/arXiv.2504.12516.

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, Wenhao Huang, Yang Wang, and Ke Wang. Widesearch: Benchmarking agentic broad info-seeking. *CoRR*, abs/2508.07999, 2025. doi: 10.48550/ARXIV.2508.07999. URL https://doi.org/10.48550/arXiv.2508.07999.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency. *CoRR*, abs/2505.22648, 2025a. doi: 10.48550/ARXIV.2505.22648. URL https://doi.org/10.48550/arXiv.2505.22648.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversation framework. *CoRR*, abs/2308.08155, 2023. doi: 10.48550/ARXIV.2308.08155. URL https://doi.org/10.48550/arXiv.2308.08155.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, and Jingren Zhou. Resum: Unlocking long-horizon search intelligence via context summarization. *CoRR*, abs/2509.13313, 2025b. doi: 10.48550/ARXIV.2509.13313. URL https://doi.org/10.48550/arXiv.2509.13313.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: agentic memory for LLM agents. *CoRR*, abs/2502.12110, 2025. doi: 10.48550/ARXIV.2502.12110. URL https://doi.org/10.48550/arXiv.2502.12110.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023a. URL http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023b. URL https://openreview.net/forum?id=WE_vluYUL-X.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. Agentfold: Long-horizon web agents with proactive context management. *CoRR*, abs/2510.24699, 2025. doi: 10.48550/ARXIV.2510.24699. URL https://doi.org/10.48550/arXiv.2510.24699.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. Aflow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=z5uVAKwmjf.

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

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 414–431. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.22. URL https://doi.org/10.18653/v1/2025.emnlp-main.22.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 62743–62767. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.

A Implementation Details

This appendix gives the configuration details that were summarized in §3.

Per-agent tool stack. The tool stack is benchmark-dependent. On BrowseComp and WideSearch each agent is given a web-search tool and a page-visit tool. On HLE we additionally provide a Python execution tool and a Google Scholar search tool, which are necessary to support the more reasoning- and literature-heavy questions in that benchmark. All multi-agent systems (Naive-, Swarm-, AgentFugue) use the identical per-agent tool stack so that comparisons are not confounded by tool capability.

Interaction budget. We match the per-query interaction budget across all multi-agent systems at 150 rounds. For Naive- and Swarm-Multi-Agent the budget is split between the meta-agent and the subagents: each subagent is capped at 100 rounds and the meta-agent at 50 rounds. For AgentFugue, which has no meta-agent, each of the N peer agents is capped at 150 rounds.

Context and hub-write trigger. Every peer agent uses a 128k context window. AgentFugue's hub-write trigger fires when an agent's running context reaches 64k tokens, at which point the agent flushes its working state into a hub episode and continues from a compressed prompt. The hub itself is initialized from Qwen3.5-9B and is trained as described in §2.3.

Evaluation protocol. For the main results we follow each benchmark's official judging protocol (LLM-as-a-judge for BrowseComp and HLE; structured field-level matching for WideSearch). The exact judges and metrics per benchmark are reproduced in Appendix B.

Hub training data. We filter OpenSeeker and OpenResearcher trajectories by removing substantively short interactions and trajectories whose final answers do not match verified gold answers, leaving 2,000 training trajectories. GPT-5.4 generates the reference episode notes and intent-conditioned readouts used for SFT. The short-trajectory cutoff is applied before teacher generation.

SFT configuration. The memory backbone is Qwen3.5-9B, updated with full-parameter supervised fine-tuning using the `ms-swift` trainer. We train for 2 epochs on 8 GPUs with per-device batch size 1 and gradient accumulation 8 (effective batch size 64), using AdamW with learning rate 1×10^{-5}, a constant learning-rate schedule, 5% linear warmup, sequence length 100K, and bfloat16 precision. The SFT objective jointly covers episode writing and intent-conditioned reading.

B Detailed Benchmark Descriptions

This appendix expands on the three benchmarks summarized in §3. For each, we describe the task format, the split we evaluate on, and the judging protocol.

BrowseComp (Wei et al., 2025). A browser-based question-answering benchmark in which each question requires multi-hop web search, cross-document evidence aggregation, and verification before a short factual answer can be committed. Questions are deliberately authored so that the answer cannot be reached by a single search query: the agent must plan, retrieve, follow links across heterogeneous web pages, and reconcile conflicting or partial evidence before answering. For evaluation efficiency and to keep the compute budget manageable, we follow prior work and evaluate on a fixed 200-question random sample of the released test set rather than the full test set, using the official LLM-as-a-judge protocol that compares the agent's extracted answer to the gold answer. For the scaling study (§3.4) we further restrict to a fixed 100-question subset (drawn from the same 200-question pool) so that runs at different team sizes N are directly comparable; the same subset is used for ablations.

WideSearch (Wong et al., 2025). A breadth-oriented web research benchmark whose tasks require collecting and consolidating many parallel pieces of evidence, for example, enumerating the attributes of a list of entities or building a structured table from many independent sources. Unlike BrowseComp, WideSearch rewards *coverage* more than depth: an agent must issue many partially independent searches, keep their results organized, and avoid omissions. We follow the official structured field-level matching protocol, which scores each predicted record against the gold record at the field level and reports the resulting accuracy. This setting tests whether peer-agent communication helps a team avoid redundant work while maintaining wide coverage.

HLE (Humanity's Last Exam) (Phan et al., 2025). A challenging closed-book / limited-tool reasoning benchmark composed of expert-authored questions across mathematics, the natural and social sciences, and the humanities. HLE evaluates capabilities that are less about web navigation and more about deliberate multi-step reasoning, allowing us to probe whether the benefit of collective reasoning transfers beyond search-heavy workloads. As with BrowseComp, for evaluation efficiency we follow prior work and evaluate on a fixed 200-question random sample of the released test set rather than the full set, using the official LLM-as-a-judge protocol with task-specific tolerance for equivalent forms.

C Detailed Baseline Descriptions

This appendix expands on the baselines summarized in §3. For each system we describe (i) its underlying model, (ii) the agentic scaffolding or training that distinguishes it, and (iii) how we instantiate it for our comparison.

C.1 LLM-based ReAct Agents

This group runs a frontier language model inside a standard ReAct (Yao et al., 2023b) loop with the same web-search and page-visit tools used by AgentFugue.

Closed-source models (Claude-Opus-4.5, Kimi-K2.5, GLM-4.7). For all three closed-source baselines, we access the model through its official API and wrap it in the same ReAct scaffold so that the comparison is about the underlying model rather than its proprietary harness. Decoding parameters are left at their official defaults.

Qwen3.5-35B-A3B. The same open-weight backbone we use inside every AgentFugue agent. We serve it locally with vLLM and call it through an OpenAI-compatible endpoint. Including Qwen3.5-35B-A3B as a single-agent baseline gives a like-for-like reference: any improvement of AgentFugue over Qwen3.5-35B-A3B (ReAct) comes from the multi-agent scaffold and the hub, not from a stronger model.

C.2 DeepResearch Agents

This group includes single-agent systems explicitly designed for long-horizon web research, typically combining a tool-using policy with task-specific scaffolding (search planning, summary memory, iterative refinement) and, in several cases, dedicated post-training on web-research trajectories.

We do *not* re-run these systems ourselves: instead, for each baseline we directly cite the numbers reported in the original paper or technical report on the corresponding benchmark, and leave the cell blank when the original work does not report a result on that benchmark.

WebSailor-32B (Li et al., 2025c). A 32B web-research agent that augments a base model with structured search-and-navigate tools and is post-trained on long-horizon trajectories to improve sustained tool use. We use the numbers reported in the original paper.

AgentFold-30B-A3B (Ye et al., 2025). A 30B sparse-activation web agent that compresses its reasoning history through a learned "folding" operation, allowing very long trajectories within a fixed context budget. We use the numbers reported in the original paper.

Tongyi-DeepResearch (Li et al., 2025a). Alibaba's Tongyi DeepResearch system, a multi-step research agent built around iterative query planning, browsing, and summarization. We use the numbers reported in the official technical report.

OpenAI DeepResearch (OpenAI, 2025). OpenAI's DeepResearch product. We use the numbers reported in OpenAI's official technical report and follow the same benchmark settings used there.

C.3 Multi-Agent Systems

The most direct comparisons for AgentFugue are alternative ways of running multiple peer agents on the same task. To isolate the coordination mechanism, all multi-agent baselines in this group share the same backbone (Qwen3.5-35B-A3B), the same per-agent context budget, and the same web-search/page-visit tools as AgentFugue. The only differences are how agents are spawned, what they communicate, and when communication happens.

Naive-Multi-Agent. A canonical *plan, parallel-search, and aggregate* pipeline. A meta-agent first reads the input question and decomposes it into K subtasks; each subtask is dispatched to an independent subagent that runs its own ReAct loop with the full tool stack and produces a written report. Once all subagents finish, the meta-agent receives their reports and synthesizes a single final answer. Subagents do not see each other's progress while they run, and the meta-agent does not modify their plans mid-execution, so all coordination is concentrated at the planning step and at the final aggregation. We use $K{=}2$ in the main results and the same Qwen3.5-35B-A3B backbone for both the meta-agent and the subagents.

Swarm-Multi-Agent. A more flexible coordination scheme that aligns with the swarm setting popularized by Kimi-K2.5 (Team, 2026). The meta-agent is given two additional tools on top of standard tool use:

- `create_subagent(identifier, system_prompt)` instantiates a specialized subagent with a custom system prompt and a stable identifier so that it can be reused across multiple tasks.
- `assign_task(identifier, task_description)` dispatches a concrete task to a previously created subagent and returns its task report.

Compared to Naive-Multi-Agent, this lets the meta-agent specialize and reuse subagents on demand, and interleave further planning with subagent invocations rather than committing to a single up-front decomposition. However, communication between subagents still happens only through the meta-agent and is mediated by final-answer-style task reports rather than by intermediate reasoning traces. We use the same Qwen3.5-35B-A3B backbone and the same per-subagent budget as the other multi-agent baselines.

D Answer Aggregation Strategies

This appendix specifies the answer-aggregation strategies referenced in §3 and analyzed in §3.4. Each strategy maps the N candidate answers $\{(a_i, c_i, t_i)\}_{i=1}^{N}$ produced by the team to a single team-level prediction, where a_i is the agent's extracted answer, c_i its self-reported confidence, and t_i its

number of tool calls. We deliberately restrict attention to lightweight, training-free aggregators that operate on top of the same set of trajectories, so that any difference between strategies reflects *how* the team's outputs are combined rather than additional compute.

Best-of-N (BoN), the default in Table 1. The team prediction is the answer of the single agent with the highest self-reported confidence, $\hat{a} = a_{i^\star}$ with $i^\star = \arg\max_i c_i$. While reward-model scores are commonly used as the weighting signal in chain-of-thought aggregation, recent work on agentic tasks has shown that an agent's own self-reported confidence is a strong and cheap alternative when an external reward model is unavailable. We adopt this convention and use BoN as the default aggregator throughout the main table.

Majority Vote (MV). A standard self-consistency-style aggregator: the team prediction is the answer that appears most frequently among $\{a_i\}$, with ties broken by self-reported confidence. MV is the natural baseline for measuring how much of the gain comes from agreement among peers rather than from any single agent's confidence calibration.

Weighted Majority Vote (WMV). A confidence-weighted refinement of MV: each candidate answer contributes a weight equal to its self-reported confidence, and the prediction is the answer with the largest total weight. WMV is meant to interpolate between BoN (which uses confidence but ignores agreement) and MV (which uses agreement but ignores confidence).

Fewest Tool Calls (FewTool). An efficiency-oriented selector: among the N candidate answers, return the one produced by the agent with the smallest number of tool calls t_i, breaking ties by self-reported confidence. The intuition is that, conditional on a correct answer, shorter trajectories are typically more reliable and cheaper to deploy; FewTool turns this heuristic into a concrete, data-free aggregator and lets us measure how much accuracy is sacrificed when the team prefers terse trajectories.

Average (Avg). Rather than aggregating into a single team answer, each agent's answer is judged independently and the per-agent correctness is averaged over the team. Avg therefore measures the expected accuracy of a *single* sampled agent and serves as a natural reference point: any aggregator that cannot beat Avg is failing to exploit the team.

Pass@k. Following the convention from code generation, a question is counted as solved if at least one of k independently sampled agents produces the correct answer. Pass@N coincides with the Oracle defined below; reporting Pass@k for $k < N$ characterizes how coverage grows with the number of parallel rollouts and isolates the contribution of sampling diversity from that of the selection rule.

E Case Studies: When Shared Memory Helps and When It Misleads

We highlight two BrowseComp runs from the same 3-agent AgentFugue configuration to make the role of the shared page memory concrete: one in which memory genuinely accelerates a downstream agent's exploration (§E.1), and one in which the same mechanism induces a confirmation bias that overrides hard constraints (§E.2). Both runs use the `multi_agent_react_with_mem_v2` hub, which compresses each agent's evicted live context into shared *pages*; downstream agents receive page summaries in their prompt and may issue `memory(pages=[...], goal=...)` calls to recover raw content on demand.

E.1 Success Case: Memory as a "Failure Map"

Question and outcome. The puzzle asks for the founding year of a 19th-century Shanghai store; the gold answer is `1853`. In this run, Agent-1 ends at `1885` and Agent-2 at `1848`; only Agent-0 reaches `1853`. Crucially, Agent-0 never directly copies a teammate answer. In fact, the teammate pages it consults explicitly state "store still unidentified."

What memory contributed. Agent-1's Page 2, which Agent-0 retrieves at step 34 with the goal "find details about the 19th-century store in eastern Shanghai . . . and the exact founding year," returns not an answer but a *failure map*: a list of candidates (Sincere, Wing On, Sun Sun, Lane Crawford, Whiteaway Laidlaw, Hall & Holtz) together with the precise reason each was rejected ("too late," "no YSB connection," "no founder-from-Canton fit"), plus an explicit `open questions` block stating that the store itself remains unidentified and that an Eastern-Shanghai store dealing in *foreign cloth* is the right direction to pursue. The shape of the returned content is shown in Box E.1.

Memory call at Agent-0 step 34

```
TOOL: memory
ARGS: { "pages": [2], "goal": "Find details about the 19th-century
store in eastern Shanghai, the manager from Chang'an/Xi'an who joined
Yokohama Specie Bank, and the exact founding year" }
MEMORY RESULT (excerpts):
- Strongly inferred bank: Yokohama Specie Bank (YSB).
- Founder's home city likely Guangzhou/Canton; manager's home city
likely Chang'an/Xi'an.
- Store location inference: "Eastern Shanghai" + "one of the first to
deal in a foreign product" points to Nanjing Road / East Nanjing Road.
- Candidates examined: Sincere (1900, too late); Wing On (1907, too
late); Sun Sun (1926, too late);
  Lane Crawford (1850 HK, no YSB connection); Whiteaway Laidlaw (1882
Calcutta, no Canton fit);
  Hall & Holtz (strongest Shanghai candidate so far, but founding year
conflicting and no manager/YSB link).
- Manager clue: not identified; no source linking a Xi'an-born
manager to a Shanghai store and YSB.
- Open question: the store itself remains unidentified.
```

How the downstream agent uses it. Steps 35–36 of Agent-0 demonstrate the intended use of this failure map. The agent first follows up on the teammate's strongest hypothesis (Hall & Holtz / Yokohama Specie Bank), confirms that this lead also dead-ends, and *then*, rather than restarting from scratch, reuses the teammate's narrowed framing ("Eastern Gate, foreign cloth, 1850s") to issue a new query (Box E.1). This query surfaces a CEFC academic PDF, which the agent visits at step 37 and which returns the decisive evidence: "*One of these stores was Dafeng created in 1853 by Weng Nianfeng . . . .*" Agent-0 then cross-checks remaining mismatches in clue interpretation across steps 38–43 (consulting Page 3 of Agent-2, which again returns "not identified") before committing to `1853`.

Agent-0 step 36 pivot after the failure-map memory call

```
REASONING (excerpt):
The teammate's exploration shows that they narrowed it down to
Yokohama Specie Bank, with founder from Canton, manager from Xi'an,
store on Nanjing Road.  ...
Hall & Holtz line is stalling -- let me try the foreign-cloth
direction the teammate flagged but did not run.
TOOL: search
ARGS: { "query": [
   "Yu Zhenxiang manager Yokohama Specie Bank Shanghai 1899",
   "Hall Holtz manager from Xi'an Chang'an joined bank",
   "Dafeng store Shanghai 1850s first foreign cloth founding year"
   ] }
```

Take-away. What the shared memory transports here is *process-level* state, including which directions have been ruled out and why, and which sub-problems are still open, rather than answer content. This is exactly the regime in which we expect collective reasoning to dominate independent rollouts: a single agent would have spent its remaining budget re-litigating the same dead ends.

E.2 Failure Case: Memory-Induced Confirmation Bias

Question and outcome. The puzzle is a conjunction of eight constraints (built in 1800s; co-located with a university whose 2013 to 2015 enrollment is 75k to 80k; used as a prison during two wars; preservation contributor whose father was faculty at that university; city population 100k to 125k in 2012 to 2016; etc.). The gold answer is `Fort Henry`. The team issues 10 explicit `memory` calls, among the most of any question in the run, yet still outputs the wrong answer: `Texas Prison System Central State Farm Main Building, Sugar Land, TX`.

What memory actually contained. The failure is *not* caused by missing evidence. Agent-2's Page 4 begins with "`No perfect match found yet`" and contains an explicit rejection ledger; a memory call at step 53 returns the authoritative Texas A&M enrollment figures (`2013: 53,219; 2014: 62,137; 2015: 64,326`), all outside 75k to 80k; a later call at step 72 returns Sugar Land population figures (`2015: 86,972; 2016: 87,367`), outside 100k to 125k, together with the verdict "`Gieseke/Central State Farm likely does not fit`" and a list of which numbered criteria fail. Box E.2 reproduces the relevant excerpt of the step-72 memory return.

Memory call at Agent-2 step 72

```
TOOL: memory
ARGS: { "pages": [4], "goal": "Find any other prison site with a
preservation contributor whose father was faculty at a university,
similar to the Gieseke clue" }
MEMORY RESULT (excerpts):
- Site: Texas Prison System Central State Farm Main Building, Sugar
Land, TX.
- Contributor: Bertram Gieseke; Father: F.E. Gieseke (Texas A&M
faculty 1886-1939).
- This is the only clearly established "father was faculty at a
university" match found so far.
Why Gieseke / Central State Farm likely does NOT fit the full
8-criteria puzzle:
- Sugar Land population: 2015 = 86,972; 2016 = 87,367; 2020 =
111,040.
- Fails criterion 7 (needs 100k-125k in 2012-2016).
- Fails criterion 8 (5-year increase ≈ 24,000, not 5,000-10,000).
- No clear evidence found that Central State Farm was used as a prison
during two wars.
```

How memory induced the error. Alongside this disqualifying evidence, the same pages mark the Gieseke / Central State Farm lead as "`the only clearly confirmed match for criterion 5+6`," a phrasing repeated across pages because no other "father was faculty at the university" candidate was ever found. Two amplifying effects follow. First, the *retrieval goal* at step 72 ("find any other prison site with a preservation contributor whose father was faculty at a university, similar to the Gieseke clue") reframes the entire task as "find another Gieseke," so a null result is read as evidence *for* Gieseke rather than against the path. Second, the natural-language summaries record hard failures (`fails criterion 7`, `fails criterion 8`) but do not gate the final answer on them. By step 74 the agent's reasoning has converted "`likely does not fit`" into "`closest match`" and rewrites each failed constraint into hedged near-satisfaction, as shown in Box E.2.

Agent-2 step 74 final reasoning

```
The memory confirms that the Gieseke/Central State Farm is the only
confirmed "father was faculty" match, but it doesn't satisfy all the
criteria. ...Given the very specific Gieseke clue ...I think the
answer must be related to this, even if some criteria don't perfectly
match.
1. Built in 1800s -- convict labor began late 1870s (could count as
```

1800s, even if main building was 1932).
2. City has a university -- *Sugar Land doesn't have a university, but Texas A&M is nearby (∼30 mi).*
3. Enrollment 75k-80k -- Wikipedia showed 74,829 / 77,491 (*close to range*).
4. Two wars -- *unclear but suggested.*
7. City population 100k-125k -- "*some sources say 111k-118k, official city says 84k-87k (conflicting)*".
8. 5-year increase 5k-10k -- "*conflicting.*"

Exact Answer: **Texas Prison System Central State Farm Main Building (Central Unit), Sugar Land, Texas**
Confidence: 65% *(gold answer: Fort Henry)*

Take-away. Shared memory faithfully recorded both the unique positive lead and the disqualifying evidence, but the compressed summary made the local uniqueness of one clue more salient than the conjunction of hard constraints; consulting downstream agents inherited and amplified this anchor. This suggests that text-only page memory is not by itself sufficient for multi-constraint tasks: useful refinements would include structured candidate states (`ACTIVE / RULED_OUT / HARD_FAIL`) that propagate across pages, and a final-answer gate that blocks any candidate carrying a recorded hard failure regardless of how strong its positive evidence appears.

F KEY PROMPTS

This appendix reproduces the prompts that drive the shared reasoning hub in AgentFugue. They define how raw agent context is compressed into page summaries, how a peer agent later consults those pages with an explicit goal, and what the `memory` tool looks like from the calling agent's perspective. Verbatim text is shown in monospace; only whitespace has been normalized.

Memory Manager: Page-Summary System Prompt (`_BULLET_SYSTEM`)

```
You are a memory manager for a research agent. Your job is to
compress the prior conversation and tool-use history into a concise
working memory that helps the next agent continue the task without
rereading the full transcript.
Write a factual summary of what has already been explored, tried,
confirmed, and left unresolved. Preserve only information that is
useful for continuing the work. Omit chit-chat, stylistic details,
and repeated content unless it affects the task.
Your summary should prioritize:
1. The user's goal, constraints, and preferences.
2. Key facts established during the conversation.
3. Tools used and the most important results from them.
4. Partial conclusions, promising leads, and failed approaches.
5. Open questions, uncertainties, and what still needs to be done
   next.
When relevant, include: filenames, URLs, document names, entities,
dates, and parameters already examined; specific findings from tool
outputs; decisions already made and why; unresolved blockers or
ambiguities.
Requirements: be concise but information-dense; be factual and do
not invent details; distinguish clearly between confirmed findings
and tentative inferences; focus on continuation value; avoid full
sentences when bullets are more efficient; do not address the user; do
not add preamble or commentary; output only the summary.
```

Window-to-Page User Template (_WINDOW_SUMMARY_USER_TEMPLATE)

```
Previous conversation and tool-use history:
{window_content}
Summarize it for continuation.  Output only the summary.
```

Page Consult: System Prompt (_CONSULT_SYSTEM)

```
You are a research assistant.  Given a research goal and retrieved
pages from past explorations, extract the information that is relevant
to the goal and produce a concise, focused summary.
Rules:
1. Keep only information that is directly relevant to the research
   goal.  Preserve important facts, findings, dates, names, and
   evidence when present.
2. Incorporate prior extracted results when provided.  Do not drop
   previously established key information unless it is contradicted or
   irrelevant.
3. Add important new information from the current page, while avoiding
   repetition.
4. Distinguish clearly between confirmed information and uncertain or
   incomplete information.
5. Be concise, factual, and information-dense.
6. Output only the extracted information and summary.
```

Page Consult: Incremental User Template (_CONSULT_INCREMENTAL_USER_TEMPLATE)

```
Research goal:
{goal}
Previous extracted results:
{previous_summary}
Current page:
{page_content}
Integrate the previous results with the current page, keeping only
information relevant to the goal.  Output only the updated extracted
information and summary.
```

memory Tool Description Shown to the Calling Agent

```
Recall and summarize the raw content of relevant past explorations
from your memory pages.  Use this tool in the following cases:
(1) you need to call memory to recover the detailed content from
those pages, but earlier conversation turns are no longer available
in the current context; (2) you need to look up specific details
and verify or cross-check the content stored in the Exploration
Memory.  First read the bullet-point summaries under each page in
the Exploration Memory section, then choose ONLY the pages whose
exploration directions are directly relevant to your current goal.
Parameters.  pages:  array of integer page numbers to retrieve (max
5); read the per-page summaries in Exploration Memory and select
only the relevant ones.  goal:  the specific information you want
to extract from these pages.
```

G Limitations and Broader Impact

Limitations. This work studies whether scaling out peer agents on the same task can yield capability gains through collective reasoning. While the current results are promising, several limitations remain.

First, our current implementation instantiates the shared reasoning hub with a moderate-sized language model and studies a limited set of agent backbones and configurations. We do not yet evaluate

the full space of stronger base models, alternative model families, or larger hub capacities, so the degree to which the observed gains transfer across scales remains an open question.

Second, the empirical study focuses on challenging long-horizon reasoning benchmarks, but does not yet cover broader settings such as open-ended report writing, sustained software engineering, or real-world interactive workflows with richer tool ecosystems. We believe the AgentFugue framework is extensible to such settings, but its behavior there remains to be validated.

Third, collective reasoning introduces its own failure modes. If episode notes are low quality, incomplete, or overconfident, the shared hub may propagate misleading intermediate conclusions across the team. Likewise, if many agents repeatedly read similar high-salience notes, communication can reduce trajectory diversity and lead to premature convergence. Better confidence calibration, diversity-aware reading policies, and more adaptive note selection remain important directions for future work.

Broader Impact. The central contribution of this paper is the idea that agent capability can scale not only by making a single agent stronger, but also by enabling teams of peer agents to share intermediate reasoning during search. This perspective may be useful for building more effective systems for knowledge-intensive tasks such as scientific assistance, open-domain research, investigative analysis, and other settings where different exploratory paths can productively inform one another.

At the same time, stronger multi-agent reasoning systems may also amplify misuse. Systems that can coordinate evidence gathering, synthesize partial discoveries, and scale out across many agents could be applied to high-volume surveillance, strategic manipulation, or more efficient generation of deceptive or misleading content. In addition, if a shared reasoning hub propagates erroneous intermediate conclusions, those errors may spread across the whole team rather than remaining isolated to a single trajectory. These risks suggest that future deployments should consider safeguards such as access control, usage monitoring, confidence-aware hub outputs, and mechanisms that preserve diversity rather than over-synchronizing agent behavior.

H Ablation: Hub Context-Window Size

This appendix accompanies the ablation discussion in §3.6. We sweep the per-agent context-window budget allocated to the shared reasoning hub at team size $N{=}2$ on the BrowseComp evaluation subset, holding every other component of AgentFugue fixed. Table 2 reports the resulting accuracy under the same five aggregator rules used in §3.4: Pass@N, MV@N, WMV@N, BoN@N, and FewTool@N (Appendix D).

Table 2: Effect of the hub context-window budget on BrowseComp at $N{=}2$. Best per column in **bold**.

Context	Pass@2	MV@2	WMV@2	BoN@2	FewTool@2
16K	49.5	47.0	47.0	47.5	45.0
32K	**60.0**	**56.0**	**58.0**	**58.0**	**53.0**
64K	52.0	42.5	50.5	50.0	44.5
96K	47.0	41.5	45.0	47.5	38.0
128K	39.0	39.0	36.5	36.0	36.5

The trend is consistent across all five aggregators: accuracy is non-monotone in the hub context budget, with a clear peak at 32K and degradation at both extremes. Very small budgets (16K) cut off useful intermediate evidence before it can be summarized into the hub; very large budgets (96K, 128K) dilute the hub's attention with stale or low-utility content and trigger memory pressure that interferes with continued exploration. Notably, the 32K configuration outperforms the 64K setting deployed in our main experiments by a substantial margin under every aggregator (e.g., $+8.0$ Pass@2, $+13.5$ MV@2, $+8.5$ FewTool@2). This means the headline numbers reported in Table 1 are *not* the strongest results AgentFugue can deliver: they reflect a deliberately conservative deployment choice (64K hub-write trigger inside a 128K agent context) that retains long-trajectory headroom for harder

questions, rather than an upper bound on what the same method achieves once the hub budget is tuned.