nature machine intelligence

Article

https://doi.org/10.1038/s42256-026-01268-y

Capable language models can outgrow the benefits of collaboration

Received: 20 December 2025

Accepted: 4 June 2026

Published online: 24 July 2026

Check for updates

Yubin Kim[1,2] ✉, Ken Gu[1], Chanwoo Park[2], Chunjong Park[3], Samuel Schmidgall[3], A. Ali Heydari[1], Yao Yan[1], Zhihan Zhang[1], Yuchen Zhuang[3], Yun Liu[1], Mark Malhotra[1], Paul Pu Liang[2], Hae Won Park[2], Yuzhe Yang[1], Xuhai Xu[1], Yilun Du[1], Shwetak Patel[1], Tim Althoff[1], Daniel McDuff[1] ✉ & Xin Liu[1] ✉

Agents, language model-based systems that can reason, plan and act with tools to accomplish tasks, are widely deployed, yet it remains unclear when multi-agent coordination outperforms a strong single agent. Here we conduct a controlled experiment that holds task prompts, tools and compute budgets constant while varying only coordination structure and model capability. Across 260 configurations spanning six benchmarks, five architectures and three LLM families, we derive a predictive model using empirical coordination metrics. Across benchmarks, single-agent baseline performance emerges as the most robust predictor of whether coordination improves or decreases performance. In particular, we identify an empirical capability-saturation threshold beyond which additional agents are unlikely to improve performance. This threshold correctly predicts the effect of multi-agent coordination on performance in 94% of validation configurations on SWE-bench Verified and Terminal-Bench. We therefore interpret this threshold as a practical selection rule rather than a universal scaling principle. A second effect, baseline-scaled error amplification, survives cluster-robust inference ($P_{\text{robust}} = 0.030$) and supports the failure-mode taxonomy. The fitted model achieves cross-validated $R^2 = 0.373$ (0.413 with a task-grounded capability metric) and selects the best architecture in 87% of held-out configurations. These results provide a quantitative framework for within-domain architecture selection and for estimating when multi-agent coordination is likely to improve performance or add overhead.

Agents[1], language model-driven systems that operate through iterative cycles of reasoning, planning and acting, adapting their behaviour based on environmental or tool-generated feedback, have achieved strong performance in diverse applications ranging from code generation[2,3] and web browsing[4,5] to medical decision-making[6–8], finance[9], sustainability[10] and scientific discovery[11–13]. As tasks become more complex and require long-horizon environmental interaction, multi-agent systems (MAS) have gained attention as a way to decompose tasks, enable parallel exploration and add verification. Promising claims have followed, including the slogan 'More agents is all you need'[14], the proposal of collaborative scaling law[15] and demonstrations that MAS outperforms single-agent systems (SAS) on complex tasks[16,17]. A concurrent body of work, however, questions whether multi-agent coordination reliably outperforms strong SAS[18–23], leaving the conditions under which MAS provides benefits underexplored. Despite this adoption, quantitative guidance remains limited on when adding

[1]Google Research, Seattle, WA, USA. [2]Massachusetts Institute of Technology, Cambridge, MA, USA. [3]Google DeepMind, Mountain View, CA, USA. ✉e-mail: ybkim95@mit.edu; dmcduff@google.com; xliucs@google.com

agents improves performance and when it degrades performance. This leaves architecture selection largely heuristic and makes it difficult to decide when multi-agent coordination offers value over simpler single-agent alternatives. A comprehensive discussion of related work on multi-agent and SAS, agentic benchmarks and coordination scaling is provided in Supplementary Section G.

To determine when multi-agent coordination provides benefit, we first establish which task categories require agentic capabilities. A necessary prerequisite is distinguishing between agentic and non-agentic evaluation paradigms. Expanding from the Agentic Benchmark Checklist (ABC) introduced in ref. 24, we characterize agentic tasks as those requiring: (1) sustained multi-step interactions with an external environment, (2) iterative information gathering under partial observability and (3) adaptive strategy refinement based on environmental feedback.

These characteristics differentiate tasks such as web browsing[4], financial trading[9], software engineering[25] and interactive planning[26] from traditional static benchmarks, tasks solvable through single-shot reasoning without environmental feedback, which lack external environments, are fully observed or require identical solution strategies[27,28]. This distinction matters because, while recent agentic benchmarks have emerged (for example, SWE-bench[25], τ^2-Bench[29] and Terminal-Bench[30]), MAS evaluations have been conducted predominantly on non-agentic tasks, potentially providing misleading guidance about when collaboration provides value. This distinction is practically consequential: although LLMs achieve high accuracy on isolated code generation tasks such as HumanEval[31], real-world deployment requires agentic capabilities such as iterative debugging, repository navigation and adaptive strategy refinement as exemplified by interactive coding assistants (for example, Cursor or Copilot Workspace). MAS that show monotonic improvement with team size on static benchmarks (reaching 89% on HumanEval with five agents) exhibit fundamentally different scaling behaviour when evaluated on tasks requiring sustained environmental interaction, where coordination overhead and error propagation dynamics dominate.

This distinction reflects a trade-off between context integration and diversity[16,32]. SAS maximize context integration by maintaining a unified memory stream in which all reasoning steps share access to prior history. MAS impose information fragmentation[18]. Parallel agents enable diverse exploration, but they incur a coordination tax in which the global context must be compressed into interagent messages. This lossy communication adds synchronization overhead and cognitive load[33], changing the scaling behaviour of collaboration.

The underlying dynamics explain this discrepancy: on agentic tasks, coordination overhead scales with interaction depth, agents operate on progressively divergent world states, and errors cascade through execution chains rather than being corrected through voting. Recent work has identified cases where single strong models match or exceed MAS[21], yet the evaluation literature provides limited guidance on what factors determine collaborative success, whether semantic diversity predicts team performance, how architectural choices shape coordination costs or whether agents can detect and correct failures in extended interactions.

Rapid progress in frontier-model capabilities further complicates this question. As base LLMs gain longer context windows, better tool use and improved self-reflection, the added value of multi-agent collaboration becomes less clear. The answer probably depends on task characteristics and architectural choices that have not yet been systematically quantified.

Two issues make controlled MAS evaluation difficult. First, existing MAS evaluations compare architectures using different prompts, tools or computational budgets, conflating architectural effects with implementation choices and precluding clean causal attribution. Second, evaluations focus exclusively on final accuracy metrics without examining process dynamics such as coordination overhead, error propagation and information flow that determine whether collaboration succeeds or fails. We know from human team performance[34,35] that team effectiveness depends on composition, coordination mechanisms and member differentiation. Yet we lack comparable empirical understanding of how these principles translate to artificial agents, leaving practitioners without quantitative guidance for architecture selection.

To address these challenges, we present a controlled evaluation of agent coordination. Our experimental design isolates architectural effects by controlling for implementation confounds. We maintain identical task prompts, tools and computational budgets across all configurations and systematically vary only coordination structure and model capability. We evaluate five canonical architectures: SAS and four multi-agent variants (independent, centralized, decentralized and hybrid) instantiated across three major LLM families (OpenAI, Google and Anthropic) sampling models at varying capability tiers as quantified by an aggregate intelligence index (Supplementary Section A), on six agentic benchmarks: (1) web browsing (BrowseComp-Plus[36]), (2) financial analysis (Finance Agent[37]), (3) game planning (PlanCraft[26]), (4) realistic workplace tasks (WorkBench[38]), (5) software engineering (SWE-bench Verified[25]) and (6) terminal tasks (Terminal-Bench[30]). Across $N = 260$ controlled configurations with matched compute, we fit a predictive model across tested domains quantifying how performance is associated with empirically measured coordination properties.

In contrast to prior claims that 'more agents is all you need', we find that the effectiveness of MAS depends on quantifiable trade-offs between architectural properties and task characteristics. We establish a predictive framework using a linear regression model with empirical coordination metrics and prespecified interaction terms: efficiency (success/overhead ratio), error amplification factors, message density and redundancy as predictors, achieving cross-validated $R^2 = 0.373$ across all six benchmarks ($R^2 = 0.413$ with a task-grounded capability metric), without dataset-specific parameters. Critically, the framework selects the best-performing architecture in 87% of held-out within-domain configurations. Relative architecture selection is therefore more stable than absolute cross-domain performance prediction.

Our analysis identifies one predictor that survives both cluster-robust inference and Holm–Bonferroni multiple-comparison correction, together with a second mechanistic effect that survives cluster-robust inference and supports the failure-mode taxonomy. The Holm- and cluster-robust predictor is the single-agent baseline coefficient ($P_{\text{robust}} = 0.004$; $P_{\text{Holm}} = 0.018$), showing that baseline task difficulty is the strongest robustly supported determinant of absolute agent-system performance. A separately fitted baseline-by-team-size decision rule yields an empirically derived ~45% capability-saturation threshold and, in an additional validation analysis, predicts the sign of the multi-agent gain in 94% of 16 SWE-bench Verified and Terminal-Bench model × benchmark configurations; because the underlying baseline-by-team-size interaction does not itself survive cluster-robust correction, we present the threshold as a validated selection rule rather than a coefficient-level scaling law. The cluster-robust mechanistic effect is baseline-scaled error amplification ($P_{\text{robust}} = 0.030$): architectures without centralized verification propagate errors more as baseline difficulty grows. A trace-level analysis ranks the four MAS variants consistently, from independent (amplification 17.2×) to centralized (4.4×), illustrating that validation bottlenecks intercept errors before aggregation. Beyond these two effects, three additional patterns are directionally consistent across all six benchmarks but do not survive cluster-robust correction at $\alpha = 0.05$: a tool-coordination trade-off where tool-heavy workflows incur higher coordination overhead, a near-linear relationship between aggregate model capability and performance, and an interaction between baseline difficulty and team size. We report these as descriptive directional patterns rather than confirmed effects. Performance

changes relative to single-agent baseline span +80.8% on structured financial reasoning to −70.0% on sequential planning, demonstrating that architecture-task alignment determines collaborative success. Translating these regression coefficients into selection rules yields 87% accuracy in choosing the best-performing architecture for held-out within-domain configurations.

Our primary contributions are:

- A controlled experimental template for SAS–MAS comparison. We match task prompts, tool interfaces and per-system compute ceilings across architectures and vary only coordination structure and model capability, while measuring the additional coordination overhead each multi-agent variant actually incurs rather than treating interagent communication as free compute. The template spans 260 configurations across three LLM families and six diverse agentic benchmarks, enabling controlled attribution of performance differences to coordination choice rather than implementation confounds.
- Single-agent baseline as the most robust determinant, with capability saturation as a validated selection rule. Across both cluster-robust inference ($P_{robust} = 0.004$, clustering on dataset, $G = 6$) and Holm–Bonferroni multiple-comparison correction ($P_{Holm} = 0.018$), the single-agent baseline coefficient is the only predictor that retains significance, identifying baseline task difficulty as the strongest robustly supported determinant of absolute agent-system performance. In the separately fitted baseline-by-team-size decision rule, baselines above approximately 45% predict zero-to-negative multi-agent gains, and in an additional validation analysis the rule matches the sign of the multi-agent gain in 94% of 16 SWE-bench Verified and Terminal-Bench model × benchmark configurations.
- A predictive model for architecture selection. A linear regression with empirical coordination metrics and prespecified interaction terms achieves cross-validated $R^2 = 0.373$ across six benchmarks ($R^2 = 0.413$ with a task-grounded capability metric). Within-domain, the model selects the best-performing architecture in 87% of held-out configurations, providing a quantitative alternative to heuristic decisions about when to use multi-agent coordination.
- Architecture-level failure-mode taxonomy. A trace-level analysis quantifies error containment across the four MAS variants, ranking them consistently from independent (amplification 17.2×) to centralized (4.4×). The interaction between single-agent baseline and trace-level error amplification survives cluster-robust correction ($P_{robust} = 0.030$), supporting validation bottlenecks as the operative mechanism.
- Descriptive coordination patterns and limits of inference. Three further patterns, namely a tool-coordination trade-off, an intelligence-index linear effect and a baseline × agents interaction, are directionally consistent across all six benchmarks but do not survive cluster-robust correction at $\alpha = 0.05$. We report them as descriptive given the small number of benchmark clusters.

Results

Main results

MAS exhibits domain-dependence with architectural variation. Figure 1 reports the aggregate scaling pattern across model intelligence, and Fig. 2 breaks this down into per-benchmark performance distributions across the five architectures defined in Table 1 and three LLM families. MAS show highly variable performance across task domains, depending on problem structure and architectural choices. On Finance Agent, MAS achieve substantial improvements: centralized reaches +80.8% (mean 0.631 versus SAS 0.349), decentralized achieves +74.5% (0.609) and hybrid reaches +73.1% (0.604), driven by opportunities for distributed financial reasoning across multiple agents. On WorkBench, MAS show minimal gains: decentralized achieves +5.6% (0.664 versus SAS 0.629), whereas centralized and hybrid both slightly underperform at −1.2%. On BrowseComp-Plus, improvements remain modest: decentralized achieves +9.2% (0.347 versus SAS 0.318), with centralized essentially flat at +0.2%. By contrast, PlanCraft exhibits degradation across all multi-agent architectures. Centralized declines to −50.3% (0.282 versus SAS 0.568), decentralized to −41.5% (0.332), hybrid to −39.1% (0.346) and independent to −70.0% (0.170). To understand this contrast between Finance Agent's gains and PlanCraft's degradation, we examined execution traces from both domains. In PlanCraft, efficient single-agent trajectories follow direct execution paths. For example, crafting a `diorite_wall`:

```
Turn 1: search("diorite_wall") → recipe: 6 diorite
in 2 × 3
Turn 2: move(diorite → crafting_grid)
Turn 3: craft → task complete
```

By contrast, centralized MAS decompose inherently sequential tasks into artificial subtasks:

`Agent 1: research recipe` (redundant, since lookup is instantaneous)
`Agent 2: check inventory` (redundant, since state is visible to all)
`Agent 3: execute crafting` (the only necessary step)

This unnecessary decomposition generates substantial coordination messages on average for tasks requiring only a few execution steps, consuming token budget on coordination rather than reasoning. Conversely, Finance Agent trajectories show when coordination can help. Single-agent execution exhibits sequential bottlenecks:

```
Turn 1: web_search("merger news") → surface results
Turn 2: edgar_search("filings") → limited depth
Turn 3–7: Sequential exploration with insufficient
breadth
```

Centralized coordination enables parallel information synthesis:

```
Agent 1: Regulatory/news analysis
Agent 2: SEC filing research
Agent 3: Operational impact assessment
Orchestrator: Synthesize multi-source findings
```

The task's natural decomposability (revenue, cost and market factors can be analysed independently) aligns with the coordination structure, yielding +80.8% improvement. These trajectory patterns indicate a mechanistic basis for domain-dependence. Coordination overhead becomes counterproductive when coordination complexity exceeds task complexity (PlanCraft) but yields substantial gains when tasks naturally decompose into parallel information streams (Finance Agent).

On SWE-bench Verified, all MAS architectures show slight degradation relative to SAS (mean 0.488): hybrid −1.3% (0.481), centralized −2.6% (0.475), decentralized −6.4% (0.456) and independent −12.8% (0.425). This is consistent with the capability-saturation threshold: most models achieve single-agent baselines above 45%, leaving limited room for coordination gains. On Terminal-Bench (SAS mean 0.312, below the threshold), results are mixed: independent shows marginal gains (+6.0%, 0.331), whereas centralized degrades substantially (−20.0%, 0.250), suggesting that the low tool count (two tools) limits the benefit of orchestration-heavy architectures.

Aggregating across all six benchmarks and architectures, the overall mean MAS improvement is 0.0% (95% confidence interval (CI) −58.7% to 77.2%), reflecting substantial performance heterogeneity with high variance ($\sigma = 37.5\%$). The performance range across MAS variants spans from −70.0% (PlanCraft Independent) to +80.8% (Finance

https://doi.org/10.1038/s42256-026-01268-y

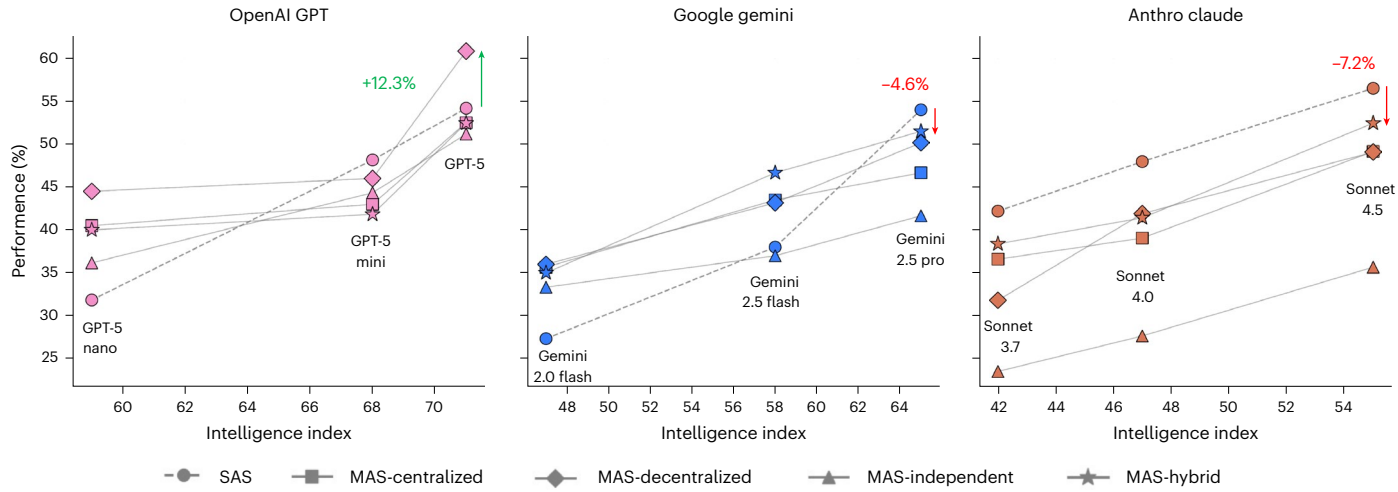


Fig. 1 | Agent scaling across model intelligence and system topologies. Average performance (%) across the six agentic benchmarks generally increases with model intelligence index across three LLM families (OpenAI, Google and Anthropic) under five agent architectures. SAS serves as reference trajectories, while MAS variants (centralized, decentralized, independent and hybrid) reveal distinct scaling behaviours (see Table 1 for architecture comparisons). Each curve aggregates accuracy across the six benchmarks at matched per-system compute, with per-benchmark sample sizes $n = 100$ (BrowseComp-Plus, PlanCraft and WorkBench), $n = 50$ (Finance Agent) and $n = 20$ (SWE-bench Verified, Terminal-Bench) per (model, architecture) cell. All percentage deltas annotated indicate the relative performance change of the best-performing MAS variant compared with the SAS baseline at the same intelligence index, averaged across the six benchmarks (per-benchmark deltas are reported in Fig. 2 and in the Results).

Centralized), indicating that MAS do not provide universal benefits but rather domain-specific trade-offs.

Task decomposability, not complexity alone, shapes coordination efficacy. Empirical patterns across benchmarks reveal that domain complexity (see Supplementary Section C for details) moderates MAS advantage. Structured, decomposable domains show large gains while high-complexity sequential domains show consistent degradation. The mechanism operates through fixed computational budgets (matched per-system compute ceilings across MAS and SAS). In structured, decomposable domains (Finance Agent, moderate WorkBench instances), agents complete local reasoning with residual capacity available for interagent communication. Here, interagent messages reduce variance through redundancy elimination and enable synthesis of partial solutions, producing large performance deltas (finance shows +80.8%). Conversely, in high-complexity sequential domains (PlanCraft), intra-agent reasoning for constraint verification and state tracking consumes most available tokens before communication can occur. Subsequent interagent messages then compress reasoning quality and produce strong negative returns (PlanCraft shows −39.0% to −70.0%).

We characterize each benchmark by a domain complexity score $D \in [0, 1]$ (Supplementary Section C), capturing the degree of sequential interdependence and empirical difficulty: WorkBench (0.000, minimal sequential constraints) shows positive MAS returns or minimal overhead, SWE-bench Verified (0.255, decomposable engineering tasks) has low domain complexity but high single-agent baselines that trigger capability saturation, Finance Agent (0.407, moderate decomposability) and Terminal-Bench (0.414, diverse CLI tasks) sit near the critical threshold, whereas PlanCraft (0.419, high sequential dependencies) and BrowseComp-Plus (0.839, dynamic state evolution) show degradation or minimal gains. Domain complexity alone does not fully predict MAS effectiveness. Although low-complexity domains (WorkBench, $D = 0.00$) show modest gains and high-complexity domains (BrowseComp-Plus, $D = 0.84$) show limited benefits, the critical factor is task decomposability: Finance Agent ($D = 0.41$) achieves +80.8% gains through parallelizable subtask structure, whereas PlanCraft ($D = 0.42$) degrades by −70% due to strict sequential dependencies despite similar complexity scores. This suggests that sequential interdependence, rather than complexity alone, determines coordination viability. Information gain $\Delta\mathcal{I}$ correlates with this pattern: Finance Agent (structured domain) exhibits strong information-value convergence ($r = 0.71$, $P < 0.001$), whereas PlanCraft (sequential constraints) shows weak correlation ($r = 0.18$, $P = 0.22$), indicating that agents in high-complexity domains exchange limited actionable information due to inherent sequential dependencies and state-space ambiguity.

Architecture-LLM family interactions differ by benchmark. Although domain complexity broadly moderates MAS effectiveness, the architecture–domain interaction shows non-uniform preferences even within similar complexity regimes. No single architecture dominates across all domains and vendors. Architecture effectiveness depends on domain structure. `Finance Agent` benefits most from centralized (+80.8%) and decentralized (+74.5%), `WorkBench` benefits most from MAS-decentralized (+5.6%) and `BrowseComp-Plus` benefits most from MAS-decentralized (+9.2%). In degrading domains, architecture selection becomes a least-worst optimization, where `PlanCraft` shows hybrid as relatively best (−39.1%) compared with MAS-centralized (−50.3%) and MAS-independent (−70.0%).

Family-specific coordination preferences emerge within improvement-positive domains. On `Finance Agent`, Anthropic's MAS-Centralized achieves +127.5% (0.636 versus 0.280 SAS), indicating conservative but stable coordination, whereas Google's MAS-centralized reaches +164.3% (0.740 versus 0.280 SAS, averaging centralized performance), consistent with a relative empirical advantage of this family under hierarchical message exchange on this benchmark. OpenAI's MAS-centralized achieves +69.9% (0.79 versus 0.465 SAS). On `WorkBench`, where multi-agent overhead is less tolerable (coordination efficiency, E_c, defined as success normalized by relative reasoning-turn count, degrades from $E_c = 0.466$ for SAS to $E_c = 0.074$ for hybrid, the largest relative drop across benchmarks), Anthropic's best variant (MAS-decentralized, +10.8%) remains superior to Google (+9.5%) and OpenAI (+8.6%), reflecting relative efficiency in managing coordination costs. On `PlanCraft`, where all variants degrade, vendor preferences flatten. Anthropic shows maximum −54.5% (MAS-hybrid 0.31 versus SAS 0.68), Google shows −25.3% (best) and OpenAI shows −32.3%, indicating that communication mechanisms cannot overcome fundamental sequential reasoning constraints. Although the precise

https://doi.org/10.1038/s42256-026-01268-y

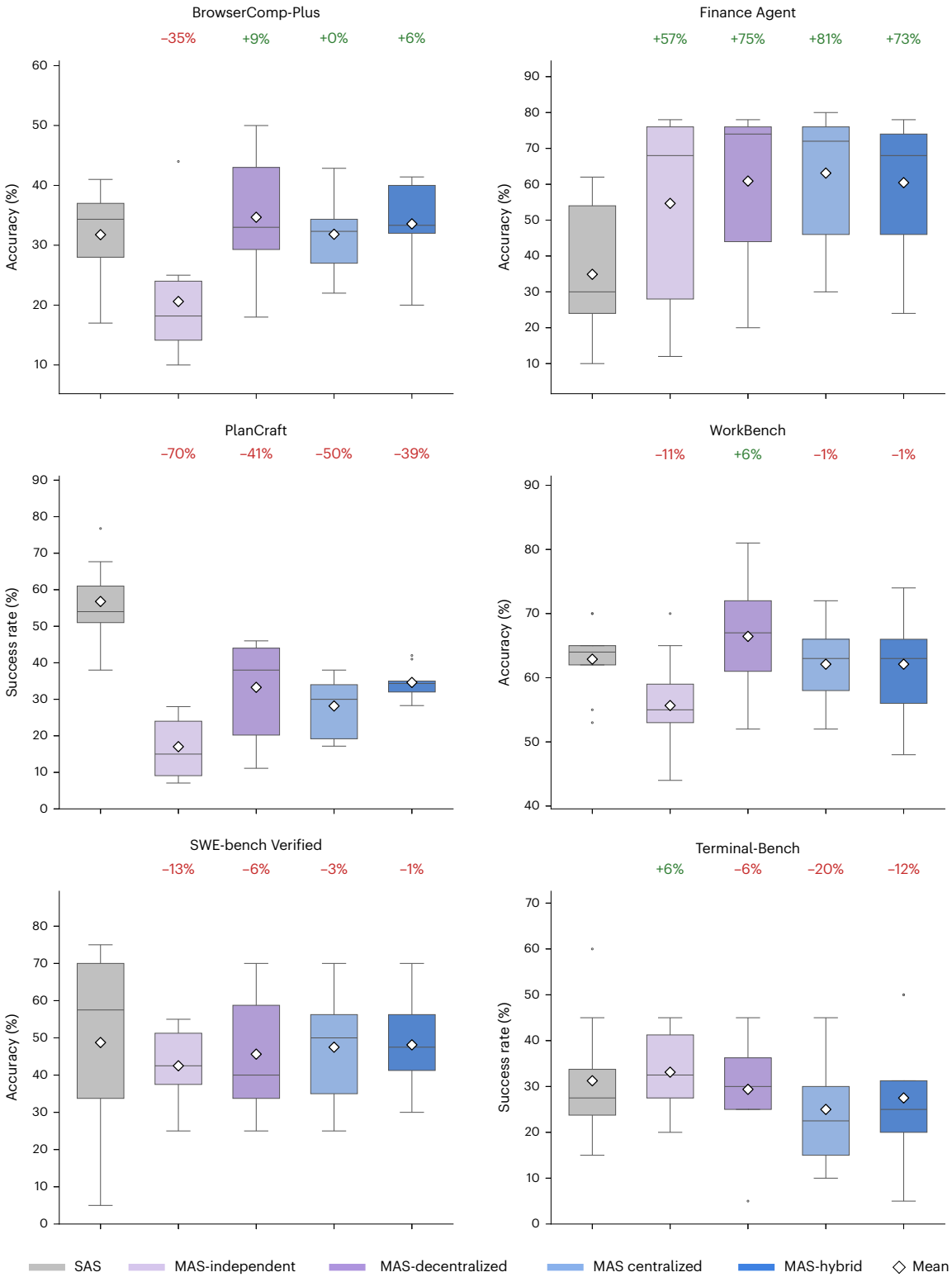


Fig. 2 | Comparative performance of agent systems across six agentic benchmarks. BrowseComp-Plus shows polarized results, with independent agents substantially underperforming relative to SAS (−35%), whereas more structured coordination achieves modest gains. Finance Agent demonstrates the strongest multi-agent benefits, with all MAS architectures substantially outperforming SAS (from +57% to +81%). PlanCraft exhibits consistent degradation across all MAS variants (from −70% to −39%). WorkBench shows marginal effects (from −11 to +6%). SWE-bench Verified shows slight degradation across all MAS architectures (from −13% to −1%), consistent with high single-agent baselines (>45%) for most models. Terminal-Bench shows mixed results, where 'independent' achieves marginal gains (+6%), whereas 'centralized' degrades (−20%), reflecting the low tool count where coordination overhead is less justified. Box plots show the distribution of model-level performance for each architecture–benchmark pair. Each model-level performance is computed over independent instances: BrowseComp-Plus, PlanCraft and WorkBench ($n = 100$), Finance Agent ($n = 50$), SWE-bench Verified and Terminal-Bench ($n = 20$). The centre line is the median, the box spans the 25th to 75th percentiles and the whiskers extend to the most extreme data points within 1.5× IQR of the box edges; points beyond the whiskers are shown individually as outliers. Whisker endpoints correspond to the lowest and highest non-outlier points. Percentage annotations represent the relative mean change versus the SAS baseline.

Table 1 | Architectural comparison of agent methods with objective complexity metrics

Characteristic	SAS	MAS (independent)	MAS (decentralized)	MAS (centralized)	MAS (hybrid)
Interaction type	Single sequential agent	Parallel agents + synthesis	Sequential peer debate + voting	Hierarchical orchestrator	Orchestrator + peer exchange
LLM calls	$O(k)$	$O(nk) + O(1)$	$O(dnk) + O(1)$	$O(rnk) + O(r)$	$O(rnk) + O(r) + O(p)$
Sequential depth	k	k	d	r	r
Common overhead	0	1	dn	rn	$rn + pm$
Parallelization factor	1	n	n	n	n
Memory complexity	$O(k)$	$O(nk)$	$O(dnk)$	$O(rnk)$	$O((r + p)nk)$
Coordination	Sequential	Parallel+synthesis	Sequential debate	Hierarchical	Hierarchical+peer
Consensus	–	Synthesis	Debate	Orchestrator	Orchestrator

Computational complexity measured in terms of LLM calls, coordination overhead and parallelization potential. Sequential depth counts the longest serialized reasoning or coordination path, and for SAS and independent, it equals the per-agent reasoning horizon k, whereas for coordinated MAS variants it counts coordination-layer rounds. *k, max iterations per agent; n, number of agents; r, orchestrator rounds; d, debate rounds; p, peer communication rounds; m, average peer requests per round. Communication overhead counts interagent message exchanges. Independent offers maximal parallelization with minimal coordination. Decentralized uses sequential debate rounds. Hybrid combines orchestrator control with directed peer communication.

mechanisms remain to be characterized, potential factors include differences in instruction-following fidelity, context utilization patterns and interturn consistency that affect how agents interpret and respond to coordination messages. No model family is consistently best across benchmarks. Each instead exhibits relative advantages in structured domains such as Finance that do not persist in sequential constraint-satisfaction domains such as `PlanCraft`, indicating that multi-agent benefits are genuinely contingent on problem structure rather than generalizable across task types.

Predictive architecture-selection model

Figure 1 summarizes the family- and architecture-level scaling patterns that motivate the predictive model developed below, and Table 1 formalizes the five canonical architectures compared throughout the study. Extended Data Fig. 1 reports cost–performance trade-offs across these configurations; Extended Data Fig. 2 reports heterogeneous-agent effects on BrowseComp-Plus; Extended Data Table 1 reports the progressive scaling-model comparison; and Extended Data Table 2 reports the coordination metrics used as inputs to the predictive model.

The main results show substantial heterogeneity, with agentic system performance ranging from +80.8% improvement to −70% degradation depending on task structure and coordination architecture. This variance correlates with measurable properties such as task decomposability, tool complexity and baseline difficulty. We next test whether measurable task, model and coordination properties can predict system performance.

Linear regression with prespecified interactions achieves cross-validated $R^2 = 0.413$ (ACI)/$R^2 = 0.373$ (intelligence index). We fit a predictive model to all 260 configurations across six benchmarks that relates agentic system performance to four categories of predictors. The first category is base model capability, captured by an intelligence index I. The second is system configuration, captured by agent count n_a. The third is task properties, captured by tool count T and single-agent baseline P_{SA}. These first three are instance-level predictors capturing within-benchmark variation, distinct from the benchmark-level domain complexity D defined in Supplementary Section C. The fourth category is empirically measured coordination metrics from Extended Data Table 2, namely efficiency E_c, overhead $O\%$, trace-level error amplification A_e^{trace}, message density c and redundancy R. Rather than including all possible terms, we construct the model based on specific mechanistic hypotheses.

Main effects capture direct relationships between individual factors and performance. We include a quadratic term (I^2) to test for nonlinear capability scaling and log-transformed tool count and agent count following standard diminishing-returns assumptions in scaling analyses[39].

Interaction terms test specific hypotheses about how these factors combine. We include nine interactions, each motivated by observed patterns: $E_c \times T$ tests whether efficiency penalties compound with tool complexity; $A_e^{\text{trace}} \times T$ tests whether errors propagate more severely in tool-rich environments; $P_{SA} \times \log(1 + n_a)$ captures the baseline paradox where high single-agent performance leaves less room for coordination gains; $O\% \times T$ tests whether overhead costs scale with task complexity. We deliberately exclude interactions without clear mechanistic justification (for example, $R \times c$, $I \times O\%$) to avoid overfitting.

The complete functional form is:

$$\begin{aligned} P = {} & \beta_0 + \beta_1(I - \bar{I}) + \beta_2(I - \bar{I})^2 + \beta_3 \log(1 + T) + \beta_4 \log(1 + n_a) \\ & + \beta_5 \log(1 + O\%) + \beta_6 c + \beta_7 R + \beta_8 E_c + \beta_9 \log(1 + A_e^{\text{trace}}) \\ & + \beta_{10} P_{SA} + \beta_{11}(I \times E_c) + \beta_{12}(A_e^{\text{trace}} \times P_{SA}) \\ & + \beta_{13}(O\% \times T) + \beta_{14}(R \times n_a) + \beta_{15}(c \times I) \\ & + \beta_{16}(E_c \times T) + \beta_{17}(P_{SA} \times \log(1 + n_a)) \\ & + \beta_{18}(I \times \log(1 + T)) + \beta_{19}(A_e^{\text{trace}} \times T) + \varepsilon, \end{aligned} \tag{1}$$

where all predictors are standardized ($\mu = 0, \sigma = 1$) after transformation. log transformations are applied to right-skewed variables spanning multiple orders of magnitude ($O\%$: 0–515%; T: 2–16; n_a: 1–4; A_e^{trace}: 1.0–17.2) to improve approximate linearity and reduce skewness. The $A_e^{\text{trace}} \times T$ interaction retains A_e^{trace} without additional log transformation because $\log(1 + A_e^{\text{trace}})$ already appears as a main effect; including $\log(1 + A_e^{\text{trace}}) \times T$ would introduce near-collinearity (variance inflation factor, VIF > 8, indicating substantial multicollinearity). Sensitivity analysis confirms qualitatively consistent results under alternative specifications ($\Delta R^2_{CV} < 0.01$). We validate model complexity through five-fold cross-validation with experiment-level holdout (splitting at the configuration level). Using the intelligence index as the capability metric, the model achieves $R^2_{\text{train}} = 0.463$, $R^2_{CV} = 0.373$ (±0.170 s.d.). Replacing the intelligence index with the task-grounded agentic capability index (ACI), defined as each model's mean single-agent performance across all six benchmarks (correlation with intelligence index: $r = 0.45$), improves model fit to $R^2_{\text{train}} = 0.481$, $R^2_{CV} = 0.413$ (±0.130 s.d.), AIC of −244.8, with no reversals in which predictors meet the pre-specified P-value threshold (Supplementary Table 8, ACI). We report the intelligence index specification in Table 2 for comparability with prior work and because ACI requires running the benchmarks but recommend ACI as the primary capability metric. The model consistently

Table 2 | Complete predictive-model coefficients relating performance to intelligence, task properties and empirical coordination metrics ($R^2_{train} = 0.463$, $R^2_{CV} = 0.373$, $N = 260$ configurations, AIC of −236.3)

Predictor	$\hat{\beta}$	95% CI	P	Interpretation
Main effects				
Intercept (β_0)	0.430	0.412 to 0.448	<0.001	Baseline performance
Intelligence ($I - \bar{I}$)	0.126	0.033 to 0.218	0.008	Linear capability effect
Intelligence^2 ($(I - \bar{I})^2$)	−0.000	−0.019 to 0.018	0.977[a]	Quadratic capability (not significant)
$\log(1 + T)$	0.166	0.095 to 0.236	<0.001[Section]	Tool diversity benefit
$\log(1 + n_a)$	0.040	−0.074 to 0.155	0.487[a]	Agent count effect
Single-agent baseline (P_{SA})	0.250	0.102 to 0.397	0.001[b]	Task difficulty proxy
Coordination structure				
$\log(1 + O\%)$	0.011	−0.033 to 0.056	0.611[a]	Direct overhead cost
Message density (c)	−0.013	−0.059 to 0.033	0.585[a]	Communication intensity
Redundancy (R)	0.006	−0.038 to 0.050	0.780[a]	Work overlap
Efficiency (E_c)	−0.011	−0.072 to 0.051	0.733[a]	Coordination efficiency
$\log(1 + A_e^{trace})$	0.014	−0.047 to 0.074	0.658[a]	Error amplification
Critical interactions				
$P_{SA} \times \log(1 + n_a)$	−0.236	−0.396 to −0.076	0.004	Baseline paradox
$E_c \times T$	−0.096	−0.154 to −0.037	0.002[Section]	Efficiency tools trade-off
$O\% \times T$	−0.033	−0.084 to 0.019	0.211[a]	Overhead scales with task complexity
$A_e^{trace} \times T$	0.022	−0.023 to 0.067	0.332[a]	Error propagation in tool-rich systems
$R \times n_a$	0.024	0.002 to 0.047	0.034	Redundancy benefit with scale
$I \times E_c$	−0.011	−0.060 to 0.038	0.653[a]	Capability-efficiency
$A_e^{trace} \times P_{SA}$	−0.080	−0.148 to −0.012	0.022[c]	Error-baseline
$c \times I$	−0.016	−0.059 to 0.027	0.457[a]	Communication-capability
$I \times \log(1 + T)$	−0.051	−0.113 to 0.012	0.112[a]	Capability-tools

The P values shown in this table are naive values. Confirmed effects are determined by the cluster-robust and Holm–Bonferroni results in Supplementary Tables 9 and 10, not by this table alone. Robustness markers next to selected P values summarize behaviour under the two corrections. 'Section' survives Holm–Bonferroni only. The single-agent baseline coefficient is the sole predictor carrying both cluster-robust inference and Holm–Bonferroni correction at $\alpha = 0.05$ and is, therefore, the only individual coefficient we describe as robustly supported. Using the task-grounded ACI improves fit to $R^2_{CV} = 0.413$. Intelligence is mean-centred ($\bar{I} = 57.5$) to address multicollinearity between I and I^2 (VIF reduced from 200 to 1.1). Model uses fivefold cross-validation. [a]Non-significant terms ($P > 0.05$, naive OLS). [b]Survives both cluster-robust inference and Holm–Bonferroni correction at $\alpha = 0.05$. [c]Survives cluster-robust inference only.

outperforms simpler alternatives using only architectural labels or intelligence alone, as shown in Extended Data Table 1. This equation contains no dataset-specific parameters or dataset-dependent tuning; it is intended for prediction on within-domain held-out configurations rather than reliable absolute prediction on unseen task domains (leave one dataset out (LODO) $R^2 = -2.09$; 'Robustness and sensitivity analysis' section).

Efficiency tools interaction as a directional pattern rather than a confirmed effect. The efficiency tools interaction shows the largest naive OLS effect among interaction terms ($\hat{\beta}_{E_c \times T} = -0.096$, 95% CI −0.154 to −0.037, naive $P = 0.002$). However, this predictor is inflated by pseudoreplication: tool count T varies primarily at the dataset level ($G = 6$ clusters), and under cluster-robust standard errors, the evidence for this effect is not retained ($P_{robust} = 0.205$; 'Robustness and sensitivity analysis' section). We therefore report the efficiency tools relationship as a descriptive directional pattern, consistent across all six benchmarks, rather than as a confirmed scaling effect. The direction is interpretable: SAS achieve $E_c = 0.466$ (Extended Data Table 2), whereas multi-agent architectures range from $E_c = 0.074$ (hybrid) to $E_c = 0.234$ (independent), a 2–6× efficiency penalty.

Conversely, simple tasks ($T \leq 4$) show negligible efficiency effects, explaining why multi-agent coordination can succeed on decomposable problems. This descriptive pattern is directionally consistent with the naive hypothesis being incomplete: tool-rich environments appear to amplify the coordination tax, plausibly making simpler architectures more effective in high-T regimes. We emphasize this is a descriptive directional finding, not a confirmed scaling law: a definitive test would require a larger number of benchmark clusters at varying tool counts than the $G = 6$ available here.

Error amplification varies by architecture. Extended Data Table 2 presents substantial variance in trace-level error amplification factors. The single-agent value is $A_e^{trace} = 1.0$. Centralized reaches $A_e^{trace} = 4.4$, hybrid reaches 5.1, decentralized reaches 7.8 and independent multi-agent reaches 17.2. After controlling for other coordination metrics, neither the main effect of error amplification ($\hat{\beta} = 0.014$, $P = 0.658$) nor its interaction with tool count ($\hat{\beta} = 0.022$, $P = 0.332$) reaches statistical significance. This does not contradict the cluster-robust baseline-scaled error amplification interaction reported in the robustness analysis ('Robustness and sensitivity analysis' section); we therefore interpret error propagation as a conditional mechanism that becomes most informative when scaled by baseline task difficulty, rather than as a standalone universal driver. The large architecture-level performance differences observed in Extended Data Table 2 are co-explained by efficiency (E_c) and overhead ($O\%$). Independent architecture's universal underperformance (mean success 0.370 versus 0.466 SAS) stems from the absence of interagent

communication, where each agent operates in isolation and duplicates errors without correction opportunities. This effect is subsumed by the efficiency metric ($E_c = 0.234$ for independent versus $E_c = 0.466$ for SAS).

Overhead shows a directional interaction with tool count. Multi-agent architectures incur substantial realized reasoning-turn overhead: independent (58%), decentralized (263%), centralized (285%) and hybrid (515%), corresponding to 1.6–6.2× more reasoning turns than SAS under matched per-system compute ceilings. This overhead reflects realized coordination turns and interagent communication steps, not an increase in the per-system token-budget ceiling, which is matched across architectures (see Methods for the per-system reasoning-token budget is allocated across parallel agents in MAS configurations). The fitted model indicates that this overhead interacts with tool count ($\hat{\beta}_{O\%\times T} = -0.033$, $P = 0.211$), a directional pattern that is not retained under cluster-robust inference in the six-benchmark model due to greater heterogeneity across task domains. The direction is preserved. For hybrid architecture ($O\% = 515$) on workbench ($T = 16$), overhead costs compound with tool complexity, explaining hybrid's collapse on tool-heavy benchmarks (success rate 0.452 overall, 0.621 on workbench). Empirically, workbench confirms this pattern. Decentralized (mean 0.664) outperforms centralized (0.621) despite higher overhead, due to better parallel efficiency. We note that this predictor retains significance under naive OLS on the four-benchmark subset ($\hat{\beta} = -0.162$, $P < 0.001$) and report it as a directional pattern under the more conservative six-benchmark specification.

Intelligence shows a directional OLS pattern ($\hat{\beta}_I = 0.126$, naive $P = 0.008$). After centring intelligence scores to address multicollinearity (variance inflation factor (VIF) reduced from 200 to 1.1), the linear capability effect is positive under naive ordinary least squares (OLS), with higher-capability models achieving proportionally better performance across all architectures. The quadratic term (I^2) does not meet the threshold (naive $P = 0.977$), indicating that capability scaling follows a linear rather than accelerating pattern within the tested range ($I \in [42, 71]$). Under cluster-robust inference this effect does not meet the pre-specified threshold ($P_{robust} = 0.059$; 'Robustness and sensitivity analysis' section), so we report it as a descriptive directional pattern rather than a confirmed scaling law.

Redundancy shows a small architecture-dependent pattern ($\hat{\beta}_{R\times n_a} = 0.024$, naive $P = 0.034$). Work redundancy, operationalized as the mean cosine similarity of agent output embeddings, ranges from 0.41 (centralized) to 0.50 (decentralized) for MAS (Extended Data Table 2). The fitted model identifies a weak positive interaction with agent count ($\hat{\beta}_{R\times n_a} = 0.024$, 95% CI 0.002 to 0.047, naive $P = 0.034$), suggesting redundancy may offer some error-correction benefit when more agents participate. Because predictors are standardized after transformation, this coefficient should not be interpreted by multiplying raw redundancy and raw team-size values; we therefore treat the term qualitatively. The magnitude of this directional effect is small relative to the efficiency tools interaction ($|\hat{\beta}_{E_c\times T}| = 0.096$, roughly four times larger in standardized units), indicating that any redundancy benefit cannot compensate for architectural inefficiency. The nominal P value ($P = 0.034$, near the $\alpha = 0.05$ threshold) suggests this relationship may be context-dependent, potentially stronger in error-prone domains or weaker when communication is expensive. Decentralized architecture, which exhibits highest redundancy ($R = 0.50 \pm 0.06$), achieves top performance on tool-heavy tasks (workbench success 0.664), consistent with redundancy's protective role. Yet this same architecture underperforms on planning tasks (0.282), where redundancy becomes wasteful duplication. This context-dependence aligns with the baseline paradox: redundancy helps when there is room for improvement ($P_{SA} < 0.45$) but becomes overhead when baseline is high.

The fitted model enables quantitative architecture selection. Equation (1) defines 20-parameter model used to rank candidate architectures. Given task characteristics (T, P_{SA}) and model capability (I), practitioners can compute expected performance for each architecture using empirical coordination metrics from Extended Data Table 2. Consider three task archetypes: (1) planning tasks ($T = 4$, $P_{SA} = 0.57$) favour single-agent due to baseline paradox and low tool count; (2) analysis tasks ($T = 5$, $P_{SA} = 0.35$) favour centralized multi-agent, balancing error control ($A_e^{trace} = 4.4$) with manageable overhead; (3) tool-heavy tasks ($T = 16$, $P_{SA} = 0.63$) favour decentralized multi-agent despite high overhead (263%), because parallelization and redundancy outweigh efficiency losses. Quantitatively, the decision boundary between single-agent and multi-agent is

$$P_{SA}^{*} = -\frac{\hat{\beta}_4}{\hat{\beta}_{17}} \approx \frac{0.040}{0.236} = 0.170 \quad \text{(in standardized units)},$$

corresponding to raw performance of approximately 0.45 after denormalization. This threshold, estimated from the present experiments, aligns with empirical best practices and provides a quantitative criterion for coordination structure selection that supplements heuristic guidance with a fitted model. Cross-validation on held-out configurations indicates that this rule achieves 87% correct architecture selection within the tested domains, substantially exceeding random choice (20%) or capability-only models (54%). The fitted model therefore offers a within-domain architecture-ranking rule estimated across the six benchmarks studied here, while absolute prediction on unseen domains remains limited ('Robustness and sensitivity analysis' section).

Coordination efficiency, error dynamics and information transfer

Following the Multi-Agent System Failure Taxonomy (MAST) proposed by [20], we categorize observed errors into specification, interagent misalignment and verification failures. Building on this taxonomy, we quantitatively analyse error frequency and propagation across architectures. We systematically characterized coordination efficiency, error propagation mechanisms and information transfer across all 260 experiments. All MAS and SAS configurations were run under matched per-system compute ceilings and identical tool-call access to isolate coordination effects; across trials, the mean reasoning-token budget was approximately 4,800 tokens.

Turn count follows power-law scaling with number of agents. Figure 3 reports the agent-count experiment for Gemini-2.0 Flash and Gemini-2.5 Pro on BrowseComp-Plus, illustrating how accuracy varies with team size $n_a \in \{1, 2, 3, 5, 7, 9\}$ under matched per-system compute. Total reasoning turns (reasoning–response exchanges) exhibit power-law growth with agent count

$$T = 2.72 \times (n + 0.5)^{1.724}, \quad R^2 = 0.974,$$

$$95\%\text{CI on exponent}: [1.685, 1.763], \quad p < 0.001.$$

This relationship is fit across architecture-aggregated means. Within-architecture variance remains substantial (for example, at $n = 3$ independent averages 11.4 turns versus decentralized 26.1 turns), reflecting topology-dependent communication patterns. The super-linear exponent (1.724 > 1) reflects quadratic message complexity (all-to-all potential communication) tempered by practical bandwidth limits, defining an agentic scaling regime distinct from neural network parameter scaling (for example, Kaplan et al. report $b = 0.76$ for dense models). Empirically, Hybrid systems require 6.2× more turns than SAS (44.3 versus 7.2 turns, $t(178) = 16.8$, $P < 0.001$; the t-statistic is computed on the 180-configuration full-cost-tracked subset visualized in Extended Data Fig. 1), centralized requires 3.8× (27.7 turns), and decentralized requires 3.6× (26.1 turns). Under fixed computational

https://doi.org/10.1038/s42256-026-01268-y

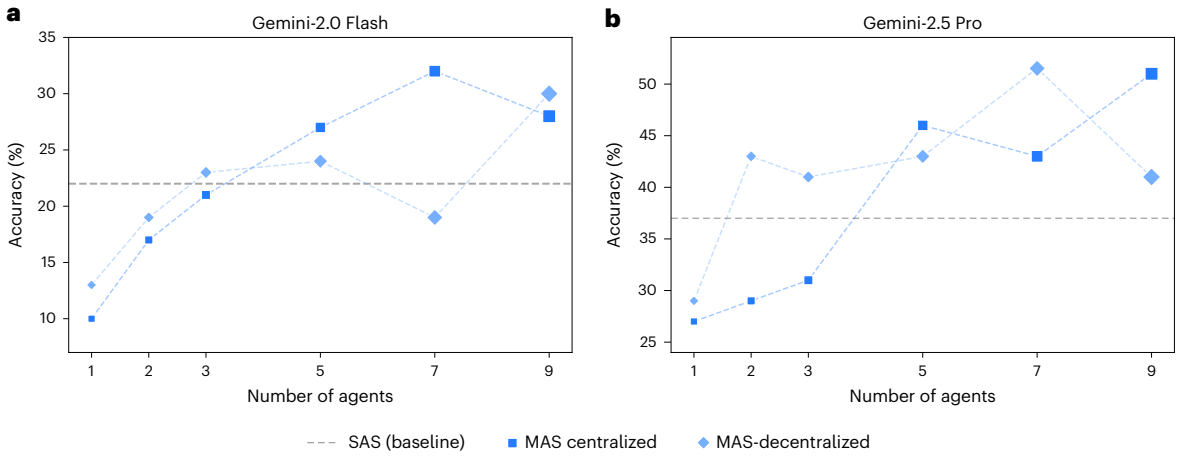


Fig. 3 | Number of agents scaling reveals model-dependent coordination limits. a,**b**, Performance of Gemini-2.0 Flash (**a**) and Gemini-2.5 Pro (**b**) on BrowseComp-Plus across multi-agent architectures with varying number of agents ($n_a \in \{1, 2, 3, 5, 7, 9\}$). Each point is the mean accuracy over $n = 100$ independent BrowseComp-Plus task instances (the same evaluation set used in the main experiments), under matched per-system compute. No error bars are shown because each cell corresponds to a single canonical cluster run. The dashed lines indicate the single-agent baseline for each model on the same benchmark. Both models show initial gains from multi-agent coordination, but scaling patterns diverge: Gemini-2.0 Flash exhibits a clear optimum at seven agents before degradation, whereas Gemini-2.5 Pro's decentralized architecture peaks earlier despite its higher single-agent baseline. The centralized architecture demonstrates more stable scaling for Flash but shows diminishing returns for Pro beyond 5 agents.

budgets, per-agent reasoning capacity therefore becomes prohibitively thin beyond three to four agents, producing an observed resource ceiling under fixed budgets where communication cost dominates reasoning capability.

Message density shows a weak descriptive trend. Success rate shows a weak positive trend with message density

$$S = 0.42 + 0.18 \ln(1 + c), \quad R^2 = 0.018, \quad p \approx 0.07,$$

where c is messages per reasoning turn; the $\ln(1 + c)$ form keeps the fit defined for the SAS and independent architectures, which have $c = 0$ by construction (no interagent communication). The two coordinating MAS architectures at $c \approx 0.4$ achieve comparable mean success rates: decentralized 48.8% ($c = 0.41$) and centralized 46.3% ($c = 0.39$). Additional total coordination expenditure does not translate into proportional gains: hybrid systems (515% coordination overhead, $T = 44.3$) achieve 47.7% success, statistically indistinguishable from centralized (285% overhead, $T = 27.7$) despite using substantially more turns and overhead ($t(70) = 0.31$, $P = 0.755$; equal-variance two-sample on $n = 36$ per architecture, drawn from the 180-configuration full-cost-tracked subset). This pattern is consistent with diminishing returns from additional coordination effort. Among high-performing runs, qualitative inspection of agent traces shows convergent content once consensus is reached, suggesting that messages beyond consensus add redundancy rather than novel information.

Error absorption mechanisms. We formalize error absorption as $\text{absorb} = (E_{SAS} - E_{MAS})/E_{SAS}$, where E is factual error rate. The absorption mechanism operates through iterative verification: in centralized and hybrid architectures, subagent outputs pass through an orchestrator that cross-checks reasoning steps before aggregation, enabling detection and correction of logical inconsistencies. In decentralized architectures, peer debate rounds provide similar verification through explicit challenge-response exchanges. These architectures achieve 22.7% average error reduction (95% CI 20.1% to 25.3%), peaking at 31.4% for Finance Agent, where structured numerical outputs facilitate verification. Independent MAS shows no error correction (+4.6% amplification) due to absence of any interagent verification mechanism where errors made by individual agents propagate directly to the aggregated output without opportunity for correction.

The correction mechanism is revealed through token-overlap analysis. Each token in agent rationales is labelled as: (1) unique (appears in exactly one agent), (2) shared (two or more agents) and (3) contradictory (semantic opposition, BERTScore <0.3). High-performing runs exhibit: (1) increased shared-token entropy (mean ~1.8 bits for Finance Agent; $P < 0.001$ versus low-performing runs); (2) substantially reduced contradictory mass (median 2.3% in successes versus 8.1% in failures), consistent with messages converging towards mutually consistent subproofs rather than self-reinforcing errors. However, high redundancy ($R > 0.50$) correlates negatively with success ($r = -0.136$, $P = 0.004$), implying an emergent diversity–efficiency trade-off in which collective capability peaks when message overlap balances shared grounding with informational diversity. Optimal redundancy occurs at $R \approx 0.41$ (the centralized median), balancing information fusion with reasoning independence.

Error taxonomy reveals architecture-specific failure modes. We identified four error categories as follows.

(1) Logical contradiction: agent asserts both 'X is true' and 'X is false' about the same entity or derives conclusions violating its stated premises; (2) numerical drift: accumulated computational error from cascading rounding or unit conversion mistakes, measured as relative deviation from ground truth exceeding 5%; (3) context omission: failure to reference previously established entities, relationships or state information required for the current reasoning step; (4) coordination failure (MAS-specific): message misinterpretation, task allocation conflicts, or state synchronization errors between agents. Architecture-specific patterns emerge across these categories:

- Logical contradiction: baseline 12.3–18.7%. Centralized reduces to 9.1% (36.4% reduction) via consensus; decentralized achieves 11.5% through peer verification; independent unchanged at 16.8%.
- Numerical drift: baseline 20.9–24.1%. Centralized/decentralized reduce to 18.3% (24% reduction) via subproblem verification; hybrid amplifies to 26.4% as rounding errors propagate; independent unchanged at 23.2%.
- Context omission: baseline 15.8–25.2%. Centralized reduces to 8.3% (66.8% reduction) via orchestrator synthesis; decentralized achieves 11.2%; independent unchanged at 24.1%.

- Coordination failure: only appears in MAS. Independent: 0% (no coordination mechanism); centralized: 1.8%; decentralized: 3.2%; hybrid: 12.4% (protocol complexity exceeds robust implementation).

These patterns identify three operational coordination regimes: (1) under-coordination ($O < 100\%$ overhead): minimal accuracy gain ($\Delta S \approx +2–4\%$), coordination mechanisms not yet engaged; (2) descriptive coordination band ($200\% < O < 300\%$ overhead): highest success–cost ratio ($E_c \approx 0.16$), dominated by centralized and decentralized, with strong error absorption; (3) over-coordination ($O > 400\%$ overhead): hybrid runs with reduced efficiency ($E_c \approx 0.11$), protocol complexity introducing coordination-failure modes. Error amplification analysis confirms: independent architectures propagate errors to 17.2× baseline (95% CI 14.3 to 20.1; no correction mechanisms), while centralized contains to 4.4× (95% CI 3.8, to 5.0) through supervised aggregation.

Information gain is associated with MAS benefit in low-complexity domains. We compute information gain $\Delta\mathcal{I}$ by comparing pre-coordination and post-coordination task-uncertainty surrogates (via Bayesian posterior variance reduction on key variables). In structured domains (Finance Agent, WorkBench), $\Delta\mathcal{I}$ correlates strongly with MAS–SAS gap ($r = 0.71$, $P < 0.001$), indicating that agents successfully exchange high-value information and synthesize it into improved solutions. In Finance Agent specifically, $\Delta\mathcal{I}$ ranges 0.8–2.1 bits (mean 1.4) for successful trials versus 0.2–0.6 bits (mean 0.4) for failures.

In open-world domains (BrowseComp-Plus), by contrast, $\Delta\mathcal{I}$ shows weak predictive power, with no $P < 0.05$ evidence, indicating that agents' messages provide limited validated information due to world ambiguity. This domain-dependent information gain pattern maps to the observed MAS benefits. Finance Agent (up to +80.8% for centralized) shows high-value information exchange, while BrowseComp-Plus (up to +9.2% for decentralized) shows world ambiguity that limits verification.

Cross-domain generalization of coordination rankings. Coordination-metric rankings, rather than absolute success levels, showed partial stability across the tested domains (Kendall $\tau = 0.89$, coefficient of variation <0.1 across architectures), indicating partial transfer of coordination patterns rather than reliable cross-domain prediction. Extrapolation to larger teams via the fitted power law predicts ~69 turns at $n = 6$ and ~157 turns at $n = 10$ (95% CI from exponent uncertainty 64 to 74 and 143 to 172, respectively), corresponding to 9.5× to 21.8× increases over the SAS baseline of 7.2 turns. This super-linear scaling reflects the observed resource ceiling under fixed budgets: beyond three to four agents, per-agent reasoning quality degrades sharply.

Economic efficiency and family-specific cost–benefit trade-offs. Token efficiency (success per 1,000 tokens) varies sharply across architectures and families. SAS achieves 67.7 successes/1,000 tokens. Centralized drops to 21.5 (3.1× worse), decentralized to 23.9 (2.8× worse) and hybrid to 13.6 (5.0× worse). Absolute dollar costs per trial vary by model. OpenAI Hybrid achieves a marginal cost of ~US$0.008 per 1% success gain (steep but tractable for structured tasks), Anthropic hybrid reaches ~US$0.024 per 1% gain (3× worse, reflecting Anthropic's higher coordination cost) and Google falls in between at ~US$0.012 per 1% gain across architectures.

Family-specific cost–performance patterns. Cross-family analysis shows distinct architectural preferences. OpenAI models show the strongest hybrid gains on structured tasks (finance: 52% success hybrid versus 39% SAS; WorkBench: 56% hybrid versus 42% SAS). Anthropic models show the most conservative and stable centralized performance (mean 43% across tasks, s.d. of 2.3%, lowest variance). Google models show consistent cross-architecture efficiency (performance range <5% across topologies). These patterns may reflect family-level differences in instruction following, context utilization, interturn consistency or other architectural and training factors that affect how models process coordination messages. We do not isolate the underlying mechanism here, so these family-specific differences should be read as empirical signatures rather than mechanistic conclusions.

Table 3 | Six agentic benchmarks used for evaluation

Benchmark	Task	Evaluation design
BrowseComp-Plus (2025)	Web browsing/information retrieval	Multi-website information location
Finance Agent (2025)	Finance	Entry-level analyst task performance
PlanCraft (2024)	Agent planning	Minecraft environment planning
WorkBench (2024)	Planning/tool selection	Common business activities
SWE-bench Verified (2024)	Software engineering	GitHub issue resolution
Terminal-Bench (2026)	CLI task execution	System admin/security/machine learning tasks

Robustness and sensitivity analysis

We subject the six-benchmark regression ($N = 260$) to three robustness checks addressing pseudoreplication, multiplicity and capability metric sensitivity.

Cluster-robust inference. We re-estimated the regression with cluster-robust standard errors (clustering on dataset, $G = 6$, CR1 correction, t_5 critical values). Predictors varying primarily at the dataset level (including `log_tools`, `efficiency_x_tools` and `baseline_x_agents`) show standard error inflation up to 2.9× relative to naive OLS estimates. We report these as directional patterns rather than confirmed effects. `single_agent_baseline` ($P = 0.004$) and `error_x_baseline` ($P = 0.030$) retain statistical support under cluster-robust inference, identifying baseline task difficulty as the most robustly supported predictor of absolute agent-system performance (Supplementary Table 9, cluster-robust); the ~45% selection threshold itself is recovered from the fitted baseline-by-team-size decision rule and is presented as an empirically validated rule, not as a cluster-robust coefficient.

Multiple-comparison correction. We evaluated 19 predictor coefficients (excluding the intercept) as a single family of simultaneous hypotheses, applying the Holm–Bonferroni step-down procedure to control the family-wise error rate at $\alpha = 0.05$. Three predictors survive correction: `log_tools` ($P_{Holm} < 0.001$), `single_agent_baseline` ($P_{Holm} = 0.018$), and `efficiency_x_tools` ($P_{Holm} = 0.026$). Two further predictors (`intelligence_centered` and `baseline_x_agents`) are suggestive ($P_{Holm} < 0.10$) and are discussed as directional patterns (Supplementary Table 10, Holm–Bonferroni). The Holm–Bonferroni correction was applied to naive OLS P values (addressing multiplicity), while the cluster-robust analysis (addressing pseudoreplication) was reported separately. Under cluster-robust inference, `single_agent_baseline` ($P = 0.004$) and `error_x_baseline` ($P = 0.030$) both retain significance at $\alpha = 0.05$. Across both correction approaches, however, `single_agent_baseline` is the only predictor that is supported by the cluster-robust analysis and the Holm–Bonferroni correction simultaneously, identifying baseline task difficulty as the single most robust determinant of absolute agent-system performance in our

study; the ~45% capability-saturation threshold itself is a separately fitted baseline-by-team-size decision rule whose underlying interaction is not cluster-robust confirmed. We retain baseline-scaled error amplification as a cluster-robust mechanistic effect that supports the architecture-level failure-mode taxonomy discussed in 'Coordination efficiency, error dynamics and information transfer' section but no longer describe it as Holm-confirmed.

Capability metric sensitivity. As an alternative to the intelligence index, we introduce the ACI, defined as each model's mean single-agent performance across all six benchmarks. The two metrics are only moderately correlated ($r = 0.45$), consistent with the concern that static benchmark composites may diverge from dynamic agentic performance. ACI improves cross-validated fit (R^2_{CV}: 0.373 → 0.413) with zero finding reversals, and we recommend ACI as the primary capability metric going forward (Supplementary Table 8, ACI).

Cross-domain generalization. LODO cross-validation with $n = 6$ folds yields mean LODO $R^2 = -2.09$. The negative LODO R^2 reflects the inherent difficulty of predicting absolute success rates across structurally diverse task domains (spanning tool counts from 2 to 16 and success rates from 5% to 80%) with a single regression surface that contains no dataset-specific parameters. However, the within-domain cross-validated model correctly identifies the optimal architecture for 87% of held-out configurations, indicating that relative performance rankings between architectures are preserved even when absolute predictions across domains are not. The empirically derived ~45% capability-saturation selection rule is evaluated separately with a 94% sign-match rate across all 16 model × benchmark configurations on SWE-bench Verified and Terminal-Bench ($P < 0.001$ by binomial test; individual comparisons have wide bootstrap 95% CIs of ±20 percentage points (pp) at $n = 20$, so this rate reflects a robust aggregate trend rather than a per-cell test).

Discussion

Although this work provides a quantitative selection rule and empirical coordination patterns for agent systems across architectures and model families, several limitations remain.

Our framework systematically compares canonical coordination structures with preliminary exploration of scaling number of agents up to nine. However, our empirical findings suggest that scaling to larger collectives may face practical barriers under the tested token-based coordination scheme: the communication overhead we measured grows superlinearly with agent count, and coordination efficiency degrades substantially beyond moderate team sizes. Whether such collectives can exhibit beneficial emergent behaviours, such as spontaneous specialization or hierarchical self-organization or whether communication bottlenecks dominate remains an open question analogous to broader questions in complex adaptive systems.

Although we explore capability heterogeneity by mixing models of different intelligence levels within the same LLM family, all agents share identical base architectures differing only in scale and role prompts. A preliminary investigation of 13 heterogeneous configurations on BrowseComp-Plus (mixing models within and across families) finds no evidence that model mixing bypasses the capability-saturation threshold: centralized heterogeneous configurations underperform their strong-model homogeneous counterparts by a mean of 12.6 pp, while decentralized configurations show marginal gains (+2.0 pp) largely attributable to the stronger constituent model (Supplementary Table 7, heterogeneous agents). Future work should investigate teams combining different model architectures, domain-specialized fine-tuning or complementary reasoning strategies to understand when epistemic diversity yields robustness rather than coordination noise. In addition, our heterogeneity experiments (Extended Data Fig. 2) hint that certain models may be better suited for orchestration versus execution roles. Systematic study of role-specialized training or selection could enable more principled team composition.

Our analysis identifies tool-heavy environments as an important descriptive failure mode for multi-agent coordination. The tool-count and efficiency-related interactions are directionally negative under naive OLS but do not survive the conservative cluster-robust framing with $G = 6$ benchmark clusters; we therefore treat them as deployment-relevant directional patterns rather than confirmed scaling laws. Developing specialized coordination protocols for tool-intensive tasks, such as explicit tool-access scheduling, capability-aware task routing or hierarchical tool delegation, represents an important direction for improving multi-agent reliability.

Although we controlled prompts to be identical across conditions for experimental validity, we did not optimize prompts specifically for each model or model family. Given known sensitivity of LLM behaviour to prompt formulation, architecture-specific prompt tuning may yield different scaling characteristics than those reported here.

Our analysis spans six agentic benchmarks, which, while diverse in task structure (deterministic tool use, quantitative reasoning, sequential planning, dynamic web navigation, software engineering and CLI tasks), may not capture the full spectrum of agentic task characteristics. The strong differentiation in MAS effectiveness across these benchmarks (Fig. 2) suggests that additional environments, particularly those with novel task structures such as embodied agents, multi-user interaction or long-horizon temporal dependencies, would further strengthen confidence in the identified thresholds and scaling principles. SWE-bench Verified and Terminal-Bench use 20-instance subsets (smaller than the 50–100 instances used for BrowseComp-Plus, Finance Agent, PlanCraft and WorkBench) due to the computational cost of Docker-based evaluation environments. Bootstrap 95% confidence intervals (10,000 resamples) yield typical widths of ±20 pp per cell. While individual pairwise comparisons are underpowered at $n = 20$, aggregate trends across eight models ×2 benchmarks remain robust (for example, the 45% threshold achieves a 94% match rate, $P < 0.001$ by binomial test). All per-cell bootstrap CIs are reported in Supplementary Table 11 (bootstrap CIs).

The economic viability of multi-agent scaling remains a practical barrier, rooted in part in the token-centric communication paradigm: current coordination requires agents to serialize reasoning into natural language tokens (or at minimum, read shared context and output agreement signals), imposing non-trivial latency and cost floors. Emerging approaches such as latent-space reasoning or direct activation sharing between models could circumvent this bottleneck, potentially altering the scaling dynamics we observe if interagent communication shifts from token exchange to more efficient representational transfer. As shown in our cost analysis ('Coordination efficiency, error dynamics and information transfer' of the Results), token consumption and latency grow substantially with agent count, often without proportional performance gains. Future work should explore efficiency-oriented designs, such as sparse communication, early-exit mechanisms or distilled coordinator models, to make multi-agent deployments economically feasible at scale. Complementary latency-oriented designs, where parallel agent branches execute speculatively and suboptimal trajectories are pruned post hoc, may trade increased total compute for reduced wall-clock time, a trade-off increasingly relevant for real-time applications where response latency dominates cost considerations. Moreover, current agentic benchmarks capture dynamic text-based environments but do not yet include long-horizon temporal dependencies or real-world feedback loops. Integrating embodied or multimodal settings (for example, robotic control, medical triage or multi-user social interaction) will test whether our observed coordination patterns generalize beyond symbolic domains.

Our regression analysis clusters observations at the dataset level ($G = 6$). With a small number of clusters, cluster-robust standard errors are known to be conservative, and several predictors that are

significant under naive OLS lose significance under cluster-robust inference (Supplementary Table 9, cluster-robust). We report both naive and cluster-robust estimates throughout, framing dataset-level predictors as descriptive patterns supported by directional consistency rather than coefficient-level confirmation under cluster-robust inference. LODO cross-validation yields negative R^2 (−2.09), confirming that predicting absolute success rates on an unseen domain is not feasible without domain-specific parameters. The within-domain cross-validated model nonetheless selects the correct architecture in 87% of held-out cases, suggesting that relative architecture rankings show partial stability across the tested domains even when absolute baselines do not. Frontier-model extrapolation in Supplementary Section B further shows that high-overhead architectures (for example, hybrid) may require architecture-specific corrections beyond the training range; the 87% architecture-selection result should therefore be interpreted as within-domain interpolation rather than unrestricted frontier-model extrapolation.

Methods

Agent system definition

Building on MAS formalism[19,24], an agent system $\mathcal{S} = (A, E, C, \Omega)$ consists of a set of agents $A = \{a_1, \ldots, a_n\}$ (where $n \geq 1$), a shared environment E, a communication topology C and an orchestration policy Ω. When $|A| = 1$, we refer to this as a SAS. When $|A| > 1$, we refer to it as a MAS. Each agent a_i perceives, reasons and acts within the shared environment via iterative feedback.

Formally, each agent a_i is defined as a tuple $S_i = (\Phi_i, \mathcal{A}_i, M_i, \pi_i)$, where:

- Φ_i is the reasoning policy (typically an LLM)
- $\mathcal{A}_i = \{\text{ToolCall}(t, \theta) : t \in \mathcal{T}, \theta \in \Theta_t\}$ is the action space consisting of tool usage, where $\mathcal{T}$ is the set of available tools (for example, web search and code execution), and Θ_t represents valid parameter configurations for tool t
- M_i is the internal memory
- $\pi_i : \mathcal{H} \to \mathcal{A}_i$ is the decision function mapping observation histories to actions

The observation history space $\mathcal{H}$ contains sequences of action–observation pairs. The decision function π_i is instantiated by the reasoning policy Φ_i (the LLM): given a history $h_{i,t}$, the LLM generates a reasoning trace and selects the next action.

For instance, a history $h_{i,t}$ = (('search(query='pandas')', 'Found 5 files'), ...) is processed by Φ_i to produce the next tool call $\alpha_{i,t+1}$.

At timestep t, agent a_i selects an action $\alpha_{i,t} \in \mathcal{A}_i$ according to:

$$\alpha_{i,t} = \pi_i(h_{i,t}), \quad o_{i,t} = E(\alpha_{i,t}), \quad h_{i,t+1} = f_i(h_{i,t}, \alpha_{i,t}, o_{i,t}),$$

where E denotes the environment, and $h_{i,0} = \{s_0\}$ contains the initial task specification. The history update function $f_i : \mathcal{H} \times \mathcal{A}_i \times \mathcal{O} \to \mathcal{H}$ appends the new action–observation pair to the agent's history: $h_{i,t+1} = f_i(h_{i,t}, \alpha_{i,t}, o_{i,t}) = h_{i,t} \oplus (\alpha_{i,t}, o_{i,t})$, subject to context window truncation when $|h_{i,t+1}| > \texttt{MAX\_TOKENS}$. This update mechanism applies uniformly to both SAS and MAS configurations. Communication between agents occurs through explicit message passing in the orchestration layer.

SAS. A SAS contains one reasoning locus ($|A| = 1$, where A is the agent set). All perception, reasoning, and action occur within a single sequential loop, producing computational complexity $O(k)$ where k is the number of reasoning iterations. SAS has zero communication overhead and minimal memory $O(k)$ but limited capacity for decomposition or verification.

MAS. A MAS is an agent system $\mathcal{S}$ with $|A| > 1$, where agents interact through communication topology C and orchestration policy Ω.

Communication topology C defines information flow patterns between agents:

- Independent: $C = \{(a_i, a_{\text{agg}}) : \forall\, i\}$ (agent-to-aggregator only, no peer communication)
- Centralized: $C = \{(a_{\text{orch}}, a_i), (a_i, a_{\text{orch}}) : \forall\, i\}$ (bidirectional star: orchestrator ↔ subagents)
- Decentralized: $C = \{(a_i, a_j) : \forall\, i, j, i \neq j\}$ (all-to-all topology)
- Hybrid: $C = C_{\text{centralized}} \cup C_{\text{peer}}$ (orchestrator plus limited peer-to-peer)

The orchestrator Ω (when present) determines: (1) how subagent outputs are aggregated (for example, majority voting and weighted synthesis), (2) whether the orchestrator can override subagent decisions, (3) whether memory persists across coordination rounds and (4) termination conditions based on consensus or quality thresholds.

MAS architectures vary by how information and control propagate among agents, creating distinct trade-offs between computation, coordination and parallelization. Table 1 formalizes these trade-offs using asymptotic notations over LLM calls, sequential depth, communication overhead and memory complexity. We selected these five architectures to form a structural ablation of coordination mechanisms:

- Independent isolates the effect of parallelism (ensemble) without communication
- Decentralized introduces peer-to-peer information fusion without hierarchy
- Centralized introduces hierarchical verification and bottleneck control
- Hybrid examines the combination of hierarchy and lateral flexibility

This design allows us to systematically attribute performance gains to specific coordination mechanics rather than generic 'multi-agent' effects. Specific configurations include:

- Independent MAS: $A = \{a_1, \ldots, a_n\}$, $\mathcal{C} = \{(a_i, a_{\text{agg}})\}$, $\Omega =$ `synthesis_only`. The `synthesis_only` policy concatenates subagent outputs without cross-validation or majority voting. The aggregator performs no analytical comparison of responses, ensuring that any performance differences arise purely from parallel exploration rather than error correction. This achieves maximal parallelization but minimal coordination, suitable for ensemble-style reasoning. For state-based Docker benchmarks (SWE-bench Verified, Terminal-Bench), the released implementation shares a single Docker container per task instance across all subagents of any given multi-agent configuration (independent included); the 'independent' label therefore refers to the absence of interagent message exchange rather than to separation of file-system or terminal state. The synthesized text is returned as `agent_output` and a deterministic insertion-order environment status (the first subagent's) is reported as `final_env_output`, without success-based selection or voting. Results on these two benchmarks should be interpreted as deterministic parallel-exploration baselines under shared environment state rather than literal state-level synthesis or fully isolated parallel execution.
- Centralized MAS: $A = \{a_{\text{orch}}, a_1, \ldots, a_n\}$, $C = \{(a_{\text{orch}}, a_i), (a_i, a_{\text{orch}}) : \forall\, i\}$ (bidirectional star), Ω = hierarchical. A single orchestrator coordinates r rounds across n subagents ($O(rnk)$). Sequential depth equals r while parallelization factor remains n. This design stabilizes reasoning but creates a bottleneck at the orchestrator.
- Decentralized MAS: $A = \{a_1, \ldots, a_n\}$, $C = \{(a_i, a_j) : \forall\, i, j, i \neq j\}$, Ω = consensus. Agents communicate in d sequential debate rounds ($O(dnk)$). Memory complexity is $O(dnk)$ as each agent stores its own debate history. The consensus threshold (0.7 in the released configuration) records threshold-approved

findings during debate; when no answer reaches that threshold, the final system output is selected by plurality/majority vote over the final-round answers with a deterministic tie-break.
- Hybrid MAS: $A = \{a_{orch}, a_1, \ldots, a_n\}$, C = star + peeredges, Ω = hierarchical + lateral. Combines orchestrated hierarchy with limited peer communication ($O((r+p)nk)$, where p is the number of peer rounds). This inherits orchestrator control while enabling lateral exchange between agents.

Communication versus coordination. We distinguish communication (message passing between agents) from coordination (strategic direction of agent activities). In centralized systems, coordination occurs through the orchestrator's task decomposition and progress monitoring, while communication involves passing findings between orchestrator and workers. In decentralized systems, communication and coordination are intertwined through debate rounds where agents both exchange information and collectively steer problem-solving direction.

Thus, SAS represents the minimal unit of agentic computation ($O(k)$), while MAS configurations explore the scaling frontier of coordination complexity, ranging from fully parallel and communication-free (independent) to fully coupled with peer consensus (decentralized). These configurations allow us to test whether performance gains arise from agent coordination and specialization or merely from increased compute through ensembling. Our taxonomy covers coordination patterns common in LLM-based agentic systems, focusing specifically on communication topology, one of several orthogonal MAS design dimensions including agent specialization[32], memory architecture and aggregation strategy. Classical coordination mechanisms such as blackboard systems assume structured message formats rather than natural language, limiting their direct applicability to LLM-based agents[19,40].

We formally define the task-level error rate as $E = 1 - P$, where P is the fraction of tasks successfully resolved. The task-level error amplification factor $A_e^{task} = E_{MAS}/E_{SAS}$ quantifies the relative error rate of a MAS compared with its single-agent baseline; $A_e^{task} > 1$ indicates that coordination introduces net errors, while $A_e^{task} < 1$ indicates net error suppression. We additionally define a trace-level error amplification factor A_e^{trace} that measures how much extra computational work arises from interagent coordination failures, estimated from execution-trace token analysis ('Coordination efficiency, error dynamics and information transfer' section of the Results). Both metrics consistently show that architectures with verification mechanisms contain errors more effectively than independent coordination, though they differ in absolute magnitude ($A_e^{task} \approx 1.1$ versus $A_e^{trace} \approx 4$) because they capture complementary aspects of error dynamics.

Agentic tasks and benchmarks

Following and extending the framework of Zhu et al. [24], we operationalize a task T as agentic when optimal performance substantially benefits from adaptive interaction. Formally, if $\tau = \{(a_t, o_t)\}_{t=0}^{T}$ represents an interaction trajectory, then

$$\frac{\max_\pi \mathbb{E}[R(\tau)] - \max_g \mathbb{E}[R(g(x))]}{\max_\pi \mathbb{E}[R(\tau)]} > \delta,$$

where π represents an interactive policy, g represents any single-forward-pass function, R measures task success, δ is a task-dependent threshold and the expectation is over task instances x and stochastic environment dynamics. This definition captures tasks where interaction provides meaningful advantage over the best possible single-shot approach.

The expected return of an optimal policy thus hinges on sequential observation-action feedback, requiring agents to gather information, plan and revise hypotheses under partial observability. Building on the Agentic Benchmark Checklist[24], we formalize three necessary properties for agentic benchmarks:

- Sequential interdependence: later actions depend on earlier observations; a one-shot policy cannot achieve high reward.
- Partial observability: critical state information is hidden and must be acquired through active querying or tool use.
- Adaptive strategy formation: the policy must update internal beliefs based on new evidence obtained through interaction.

Benchmarks lacking these conditions, such as GSM8K and MMLU, evaluate static reasoning rather than agentic capabilities. The term 'agentic' is defined relative to current model capabilities. GSM8K could in principle be posed as agentic by providing calculator tools, though current LLMs do not require such scaffolding. Conversely, tasks that are agentic today, such as SWE-bench, may become solvable via single-shot inference as models improve. Our evaluation focuses on tasks that currently require multi-step interaction for non-trivial performance.

Why environment feedback matters. Real-world deployments such as coding assistants, financial analysts and embodied robots operate under uncertainty and non-stationarity. Tasks solvable by direct prompting measure linguistic knowledge, whereas agentic benchmarks evaluate the process of intelligence: exploration, adaptation and coordination. Hence, our benchmarks are chosen such that (1) base LLMs perform poorly in single-shot mode, and (2) non-trivial performance requires multi-step environment interaction.

Benchmark design principles. Extending the framework proposed by Zhu et al. [24], we introduce additional criteria to isolate architectural effects:

- Controlled tool interface**:** identical tool APIs and observation structures for all architectures to eliminate confounds from external feedback quality.
- Controlled for parametric knowledge: within each model family, evaluation emphasizes adaptive reasoning over memorized facts. Cross-family comparisons (Results) account for inherent knowledge base differences through baseline normalization.
- Action–observation loop length**:** each benchmark enforces non-trivial trajectory length $L > 3$ to ensure sequential reasoning.
- Comparative normalization: scores are normalized to the best single-agent baseline, measuring coordination gain or loss.

Table 3 summarizes the six benchmark environments, task types and evaluation designs used throughout this study.

Setup

Benchmarks. We conducted 260 experiments across six benchmarks spanning deterministic to open-world task structures: WorkBench (deterministic code execution and tool use with objective pass/fail criteria), Finance Agent (multi-step quantitative reasoning and risk assessment), PlanCraft (spatiotemporal planning under constraints), BrowseComp-Plus (dynamic web navigation, information extraction and cross-page synthesis), SWE-bench Verified (real-world software engineering; GitHub issue resolution with seven tools including bash, file editing and test execution) and Terminal-Bench (diverse CLI tasks spanning system administration, security and machine learning training; two tools). BrowseComp-Plus, Finance Agent, PlanCraft and WorkBench each contribute 45 configurations (9 models × 5 architectures), while SWE-bench Verified and Terminal-Bench each contribute 40 configurations (8 models × 5 architectures, as Claude Sonnet 3.7 is deprecated). BrowseComp-Plus, Finance Agent, PlanCraft and WorkBench use 50–100 instances per configuration. SWE-bench Verified and Terminal-Bench use 20-instance subsets due to the computational cost of Docker-based evaluation (see Supplementary Table 11 (bootstrap CIs) for bootstrap confidence intervals). The tool-count range

spans {2, 3, 4, 5, 7, 16} across all six benchmarks. BrowseComp-Plus exhibits the highest performance variability across experimental configurations (coefficient of variation $\sigma/\mu = 0.32$ computed across all 45 BrowseComp-Plus runs spanning architectures and model families, with Anthropic models contributing substantial variance due to lower absolute performance, where σ is the standard deviation of success rates, and μ is the mean success rate). By comparison, WorkBench (coefficient of variation (CV) of 0.12), Finance Agent (CV of 0.18) and PlanCraft (CV of 0.21) show lower variability, indicating more stable performance across configurations.

LLMs and intelligence scaling. We evaluate three LLM families across multiple model sizes, spanning externally standardized intelligence index values from 42 to 71 (a composite capability score integrating reasoning, coding and knowledge benchmarks; see Supplementary Section A):

- OpenAI: GPT-5-nano, GPT-5-mini, GPT-5
- Google: Gemini-2.0 Flash, Gemini-2.5 Flash, Gemini-2.5 Pro
- Anthropic: Claude Sonnet 3.7, Claude Sonnet 4, Claude Sonnet 4.5

Claude Sonnet 3.7 was deprecated by Anthropic in February 2026 and is therefore unavailable for SWE-bench Verified and Terminal-Bench. On these two benchmarks, the Anthropic family includes Claude Sonnet 4 and Claude Sonnet 4.5, while the OpenAI and Google families remain unchanged, yielding eight models per benchmark and a total of $N = 260$ configurations. Architecture-specific scaling slopes are consistent across the three tested families. The maximum difference between any two LLM families is $\Delta_{\max} = 0.023$ (computed as $\max_{i,j}|\hat{\beta}_{\text{arch},i} - \hat{\beta}_{\text{arch},j}|$ across families $i, j \in \{\text{OpenAI, Google, Anthropic}\}$), with CV <0.02 across families. These patterns are consistent within the evaluated model range and should not be interpreted as model-agnostic mechanisms beyond it; the LODO cross-validation in 'Robustness and sensitivity analysis' section in the Results cautions against extrapolating absolute baselines outside the tested domains. To ensure computational fairness, all systems were run under matched per-system compute ceilings (matched reasoning-token budgets and identical tool-call access), with iteration counts serving as safety/termination ceilings rather than the primary fairness control. Configurations terminate earlier on task completion, consensus, or per-agent budget exhaustion. MAS configurations therefore allocate the per-system token budget across parallel agents (each receiving a smaller share of the total budget than a single agent would), whereas SAS spends the entire per-system budget within a single reasoning trajectory; iteration-level ceilings for each architecture are reported in Supplementary Section E.2.

Agent architectures and complexity. We tested five coordination topologies. The SAS is paired with four MAS variants, namely independent, centralized, decentralized and hybrid. Rather than attempting exhaustive coverage of all possible architectures, we selected these four MAS configurations to form a structured ablation over two key coordination dimensions, namely (1) orchestrator presence (hierarchical control versus flat structure) and (2) peer communication (direct subagent interaction versus isolated execution). Independent isolates pure ensemble effects without any interagent communication. Centralized introduces hierarchical verification through an orchestrator bottleneck. Decentralized enables peer-to-peer information fusion without hierarchy. Hybrid combines both mechanisms (see Table 1 for formal complexity characterization). This design enables controlled attribution of performance differences to specific coordination mechanisms rather than generic 'multi-agent' effects. Coordination complexity is parameterized by communication overhead, defined as the total number of interagent message exchanges required per task. Empirical values range from 0% (SAS) to 515% (hybrid), with independent at 58%, decentralized at 263% and centralized at 285% relative to the single-agent baseline (Extended Data Table 2); cost-performance trade-offs across these configurations are shown in Extended Data Fig 1.

Metrics and validation. Primary outcome is task success/accuracy (domain-dependent: factual correctness for Finance Agent, task completion for WorkBench, goal satisfaction for PlanCraft, page synthesis accuracy for BrowseComp-Plus). Secondary metrics include: (1) factual error rate E via domain-specific validators (Cohen's κ (ref. 41): Finance Agent $\kappa = 0.91$, WorkBench $\kappa = 0.89$, PlanCraft $\kappa = 0.87$ and BrowseComp-Plus $\kappa = 0.88$; exceeding 0.80, indicating strong inter-rater reliability); (2) information gain $\Delta\mathcal{I}$ from pre- versus post-coordination uncertainty proxies (Supplementary Section E.5); (3) token-overlap structure across agent rationales, labelling tokens as unique (appearing in exactly one agent), shared (two or more agents) or contradictory (operationalized as BERTScore[42] similarity <0.3 between assertion pairs; this 0.3 cut-off is an operational heuristic for the present analysis, not a calibrated semantic-opposition threshold); (4) efficiency metrics including success per 1,000 tokens and cost-normalized performance. All metrics are normalized per reasoning turn and per token to enable cross-architecture comparison. We select coordination metrics based on two criteria: (1) direct measurability from experimental traces without requiring ground-truth labels beyond task success, and (2) coverage of distinct aspects of coordination–performance relationships identified in prior work[20]. We excluded metrics requiring subjective human annotation (for example, solution creativity) or those exhibiting high collinearity with included measures (for example, total message count correlates $r > 0.92$ with overhead). VIF analysis confirmed no severe multicollinearity among retained predictors (all VIF <5). Specifically:

- Coordination overhead: $O = (T_{\text{MAS}} - T_{\text{SAS}})/T_{\text{SAS}} \times 100\%$: captures computational cost, identified as a primary bottleneck in production multi-agent deployments.
- Message density: c (interagent messages per reasoning turn): quantifies communication intensity, a key factor in coordination scaling.
- Redundancy rate: R (mean cosine similarity of agent output embeddings): measures agent agreement, relevant for ensemble-based error correction.
- Coordination efficiency: $E_c = S/(T/T_{\text{SAS}})$ (success normalized by relative turn count): normalizes success by cost for deployment decisions.
- Error amplification: A_e^{trace} (trace-level error propagation factor, estimated from execution-trace token analysis): quantifies how coordination failures compound through agent interactions. The complementary task-level metric $A_e^{\text{task}} = E_{\text{MAS}}/E_{\text{SAS}}$ is defined in this section.

Implementation scope and reproducibility. To clarify reproducibility, the repository[43] documents implementation-scope differences between the public normalized adapters and the upstream benchmark harnesses used for bit-for-bit parity. The public Finance-Agent and WorkBench adapters reproduce the coordination pipeline on normalized tool and grading surfaces, whereas exact upstream-score parity requires running the upstream benchmark harnesses with our coordination wrappers. The main differences concern tool availability, grading surfaces, provider-side nondeterminism, upstream benchmark versioning, Docker/runtime variability, shared-container semantics for Docker-based multi-agent runs and the metadata-only 500-character cap used by the auto-submission helper. For SWE-bench Verified and Terminal-Bench, multi-agent configurations share one Docker container per task instance. Therefore, the Independent architecture denotes the absence of inter-agent message exchange rather than isolated file-system or terminal state. For Finance-Agent and WorkBench, grading uses the full aggregated agent_output, not the capped

submit reasoning metadata field. The reproduction guide and change log enumerate these scope differences in detail.

Data availability

All benchmark datasets used in this study are publicly available: BrowseComp-Plus[36] (https://arxiv.org/abs/2508.06600, 100 instances), Finance Agent[37] (https://arxiv.org/abs/2508.00828, 50 instances), PlanCraft[26] (https://arxiv.org/abs/2412.21033, 100 instances), WorkBench[38] (https://arxiv.org/abs/2405.00823, 100-instance stratified subset of 690 upstream tasks across six domains, deterministic shuffle with seed 42), SWE-bench Verified[25] (https://www.swebench.com/, 500 instances, 20-instance subset selected via deterministic shuffle with seed 42) and Terminal-Bench[30] (https://www.tbench.ai/, 89-instance release snapshot used in our experiments, first 20 instances; the upstream benchmark may include additional tasks in later releases). Aggregate metrics for all 260 configurations are reported in the Results and in Supplementary Tables 1–11. Per-instance correctness data for SWE-bench Verified and Terminal-Bench are available via Zenodo at https://doi.org/10.5281/zenodo.20388843 (ref. 43).

Code availability

The complete experimental framework, including all coordination architectures (single-agent, independent, centralized, decentralized, hybrid), prompt templates, configuration files, analysis scripts and sanitized execution traces, is publicly available via GitHub at https://github.com/ybkim95/agent-scaling (release tag v2.1.3). The code release used for this Article (version v2.1.3) is available via Zenodo at https://doi.org/10.5281/zenodo.20388843 (ref. 43). A persistent Zenodo DOI (https://doi.org/10.5281/zenodo.20144433) resolves to the most recent archived version. The released code realizes the algorithms specified in Methods; the numerical results reported in the paper are the canonical cluster runs of these algorithms. A reproduction guide (REPRODUCTION.md) gives step-by-step instructions, including set-up scripts for downloading Finance-Agent and WorkBench from their upstream repositories without redistribution. For Finance-Agent and WorkBench, the release provides normalized adapters that reproduce the coordination pipeline on the reduced tool and grading surfaces detailed in the Methods; exact upstream parity for those two benchmarks requires running the upstream benchmark harnesses with our coordination wrappers.

References

1. Wang, L. A survey on large language model based autonomous agents. *Front. Comput. Sci.* **18**, 186345 (2024).
2. Zhang, K., Li, J., Li, G., Shi, X. & Jin, Z. CodeAgent: enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In *Proc. 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (eds Ku, L.-W., Martins, A. & Srikumar, V.) 13643–13658 (Association for Computational Linguistics, 2024).
3. Yang, J. et al. Swe-agent: Agent-computer interfaces enable automated software engineering. *Adv. Neural Inf. Process. Syst.* **37**, 50528–50652 (2024).
4. Wei, J. et al. Browsecomp: a simple yet challenging benchmark for browsing agents. Preprint at https://arxiv.org/abs/2504.12516 (2025).
5. Yao, S., Chen, H., Yang, J. & Narasimhan, K. Webshop: towards scalable real-world web interaction with grounded language agents. *Adv. Neural Inf. Process. Syst.* **35**, 20744–20757 (2022).
6. Heydari, A. A. et al. The anatomy of a personal health agent. Preprint at https://arxiv.org/abs/2508.20148 (2025).
7. McDuff, D. et al. Towards accurate differential diagnosis with large language models. *Nature* **642**, 451–457 (2025).
8. Kim, Y. et al. Mdagents: an adaptive collaboration of LLMs for medical decision-making. *Adv. Neural Inf. Process.Syst.* **37**, 79410–79452 (2024).
9. Yu, Y. et al. Finmem: a performance-enhanced LLM trading agent with layered memory and character design. *IEEE Trans. Big Data* **11**, 3443–3459 (2025).
10. Zhang, Z. et al. Sustainability assessment using multimodal artificial intelligence agents. *Nat. Electron.* https://doi.org/10.1038/s41928-026-01653-w (2026).
11. Gottweis, J. et al. Towards an AI co-scientist. Preprint at https://arxiv.org/abs/2502.18864 (2025).
12. Mitchener, L. et al. Kosmos: an AI scientist for autonomous discovery. Preprint at https://arxiv.org/abs/2511.02824 (2025).
13. Kim, Y. et al. CoDaS: AI co-data-scientist for biomarker discovery via wearable sensors. Preprint at https://arxiv.org/abs/2604.14615 (2026).
14. Li, J., Zhang, Q., Yu, Y., Fu, Q. & Ye, D. More agents is all you need. *Trans. Mach. Learn. Res.* https://openreview.net/forum?id=bgzUSZ8aeg (2024).
15. Qian, C. et al. Scaling large language model-based multi-agent collaboration. In *Thirteenth International Conference on Learning Representations* (eds Yue, Y. et al.) 41488–41505 (2025); https://openreview.net/forum?id=K3n5jPkrU6
16. Du, Y., Li, S., Torralba, A., Tenenbaum, J. B. & Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. In *Proc. 41st International Conference on Machine Learning* (eds Salakhutdinov, R. et al.) 11733–11763 (PMLR, 2024).
17. Chen, W. et al. Agentverse: facilitating multi-agent collaboration and exploring emergent behaviors. In *Twelfth International Conference on Learning Representations* (eds Kim, B. et al.) 20094–20136 (2024); https://openreview.net/forum?id=EHg5GDnyq1
18. Tran, K.-T. et al. Multi-agent collaboration mechanisms: a survey of LLMs. Preprint at https://arxiv.org/abs/2501.06322 (2025).
19. Guo, T. et al. Large language model based multi-agents: a survey of progress and challenges. In *Proc. Thirty-Third International Joint Conference on Artificial Intelligence* (ed. Larson, K.) 8048–8057 (International Joint Conferences on Artificial Intelligence Organization, 2024).
20. Cemri, M. et al. Why do multi-agent LLM systems fail? *Adv. Neural Inf. Process. Syst.* **38** (2025).
21. Gao, M. et al. Single-agent or multi-agent systems? Why not both? Preprint at https://arxiv.org/abs/2505.18286 (2025).
22. Neubig, G. Don't sleep on single-agent systems. *OpenHands* https://openhands.dev/blog/dont-sleep-on-single-agent-systems (26 September 2024).
23. Yan, W. *Don't Build Multi-agents* (12 June 2025); https://cognition.ai/blog/dont-build-multi-agents
24. Zhu, Y. et al. Establishing best practices in building rigorous agentic benchmarks. In *Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (2025); https://openreview.net/forum?id=E58HNCqoaA
25. Jimenez, C. E. et al. SWE-bench: Can language models resolve real-world github issues? In *Twelfth International Conference on Learning Representations* (eds Kim, B. et al.) 54107–54157 (2024); https://openreview.net/forum?id=VTF8yNQM66
26. Dagan, G., Keller, F. & Lascarides, A. PlanCraft: an evaluation dataset for planning with LLM agents. In *Conference on Language Modeling (COLM 2025)* 1–27 (2025).
27. Liu, X. et al. Agentbench: Evaluating LLMs as agents. In *Twelfth International Conference on Learning Representations* (eds Kim, B. et al.) 52989–53046 (2024).
28. Kapoor, S., Stroebl, B., Siegel, Z. S., Nadgir, N. & Narayanan, A. AI agents that matter. *Trans. Mach. Learn. Res.* https://openreview.net/forum?id=Zy4uFzMviZ (2025).

29. Barrès, V., Dong, H., Ray, S., Si, X. & Narasimhan, K. τ^2-bench: evaluating conversational agents in a dual-control environment. Preprint at https://arxiv.org/abs/2506.07982 (2025).
30. Merrill, M. A. et al. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *Fourteenth International Conference on Learning Representations* (2026); https://openreview.net/forum?id=a7Qa4CcHak
31. Chen, M. et al. Evaluating large language models trained on code. Preprint at https://arxiv.org/abs/2107.03374 (2021).
32. Hong, S. et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In *Twelfth International Conference on Learning Representations* (2024); https://openreview.net/forum?id=VtmBAGCN7o
33. Malone, T. W. & Crowston, K. The interdisciplinary study of coordination. *ACM Comput. Surv.* **26**, 87–119 (1994).
34. McGrath, J. E. *Social Psychology: A Brief Introduction* (Holt, Rinehart and Winston, 1964).
35. Lencioni, P. M. *The Five Dysfunctions of a Team: A Leadership Fable* (John Wiley and Sons, 2002).
36. Chen, Z. et al. BrowseComp-Plus: a fair and disentangled evaluation benchmark for deep search agents. In *Proc. 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (eds Liakata, M.et al.) 22349–22370 (Association for Computational Linguistics, 2026).
37. Bigeard, A., Nashold, L., Krishnan, R. & Wu, S. Finance agent benchmark: benchmarking LLMs on real-world financial research tasks. Preprint at https://arxiv.org/abs/2508.00828 (2025).
38. Styles, O. et al. WorkBench: a benchmark dataset for agents in a realistic workplace setting. In *Conference on Language Modeling* (2024); https://openreview.net/forum?id=4HNAwZFDcH
39. Kaplan, J. et al. Scaling laws for neural language models. Preprint at https://arxiv.org/abs/2001.08361 (2020).
40. Xi, Z. et al. The rise and potential of large language model based agents: a survey. *Sci. China Inf. Sci.* **68**, 121101 (2025).
41. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**, 37–46 (1960).
42. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q. & Artzi, Y. BERTScore: evaluating text generation with BERT. In *International Conference on Learning Representations (ICLR 2020)* (2020).
43. Kim, Y. agent-scaling: v2.1.3. *Zenodo*. https://doi.org/10.5281/zenodo.20388843 (2026).

Author contributions

Y.K., K.G. and X.L. conceived the study. Y.K., K.G., D.M. and X.L. had major involvement in designing various aspects of the study. Y.K. and K.G. had major involvement in data acquisition and ingestion. Y.K. and K.G. created initial case studies and preprocessed data. Y.K. performed experiments. Y.K. analysed results. X.L. administered the project. Chanwoo Park, Chunjong Park, S.S., A.A.H., Y. Yan, Z.Z., Y.Z., Y.L., M.M., P.P.L., H.W.P., Y. Yang, X.X., Y.D., S.P., T.A., D.M. and X.L. provided organizational support for the project. Y.K., K.G., Chanwoo Park, Chunjong Park, S.S., A.A.H., Y. Yan, Z.Z., Y.Z., Y.L., M.M., P.P.L., H.W.P., Y. Yang, X.X., Y.D., S.P., T.A., D.M. and X.L. wrote and revised the initial manuscript. All authors had access to the data and have edited, reviewed and approved the final manuscript for publication. All authors accept responsibility for the accuracy and integrity of all aspects of the research.

Funding

This study was funded by Google LLC.

Competing interests

CJ.P., S.S., A.A.H., Y. Yan, Y.Z., Y.L., M.M., Y. Yang, S.P., T.A., D.M. and X.L. are employees of Alphabet Inc. (Google LLC) and/or own Alphabet stock as part of their standard compensation package. Y.K., K.G. and Z.Z. were affiliated with Google Research as student researchers during this work, and X.X. and Y.D. were affiliated with Google Research as visiting researchers (with primary academic appointments elsewhere). Neither Y.K., K.G., Z.Z., X.X. or Y.D. are employees or stockholders of Alphabet Inc. The other authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-026-01268-y.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-026-01268-y.

Correspondence and requests for materials should be addressed to Yubin Kim, Daniel McDuff or Xin Liu.

Peer review information *Nature Machine Intelligence* thanks Lichao Sun and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2026

https://doi.org/10.1038/s42256-026-01268-y

Extended Data Table 1 | Progressive predictive-model comparison across nested specifications

Model Specification	R^2 (train)	R^2 (CV)	AIC	Number of parameters
Intelligence + Tools + Agents	0.405	0.36	-238.8	4
Intelligence + Tools + Agents + Coordination Structure	0.428	0.363	-243.2	10
Intelligence + Tools + Agents + Coordination Structure + Single-agent baseline	0.429	0.358	-242	11
Intelligence + Tools + Agents + Coordination Structure + Single-agent baseline + Interaction terms	0.463	0.373	-236.3	20

Predictive-model comparison. Progressive inclusion of empirical coordination metrics improves cross-validated predictive fit. All models use 5-fold cross-validation with experiment-level holdout (N=260, six benchmarks). The full interaction model achieves the highest cross-validated fit ($R^2_{CV} = 0.373$); AIC favors the smaller '+ Coordination structure' specification, reflecting the parameter cost of the interaction terms relative to their cross-validated gain.

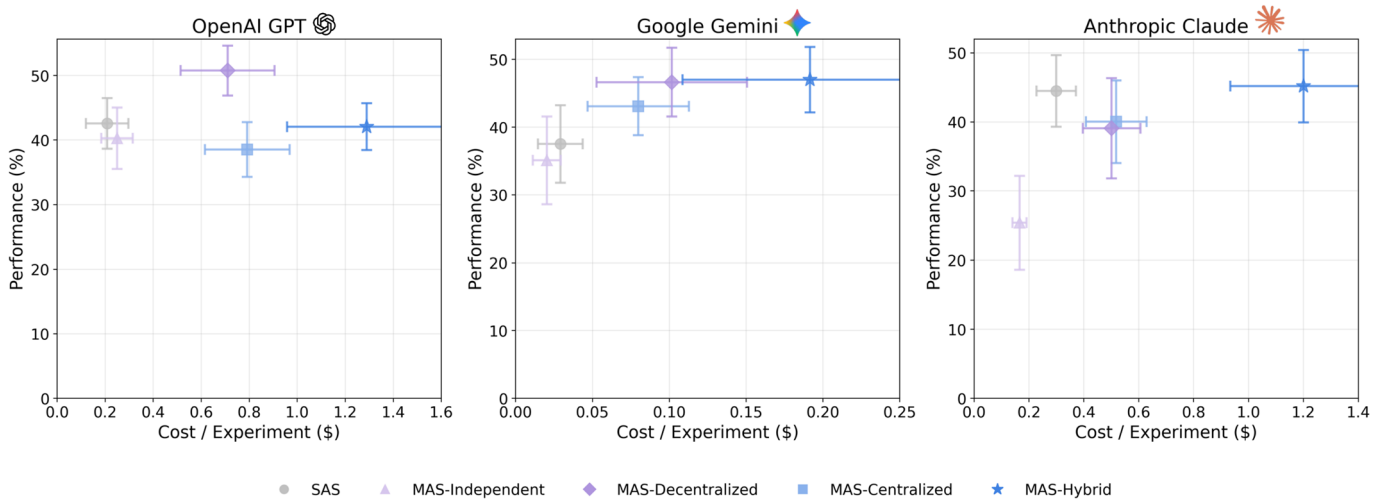
https://doi.org/10.1038/s42256-026-01268-y

Extended Data Table 2 | Coordination metrics across architectures and families

Metric	SAS	Independent	Decentralized	Centralized	Hybrid
Success Rate (S)	0.466	0.37	0.477	0.463	0.452
Turns	7.2 ± 2.1	11.4 ± 3.2	26.1 ± 7.5	27.7 ± 8.1	44.3 ± 12.4
Overhead (O%)	0	58	263	285	515
Message Density (c)	0	0	0.41	0.39	0.24
Redundancy (R)	0	0.48 ± 0.09	0.50 ± 0.06	0.41 ± 0.06	0.46 ± 0.04
Efficiency (Ec)	0.466	0.234	0.132	0.12	0.074
Error Amplification (A_e^{trace-level})	1	17.2	7.8	4.4	5.1
Success per 1K tokens	67.7	42.4	23.9	21.5	13.6

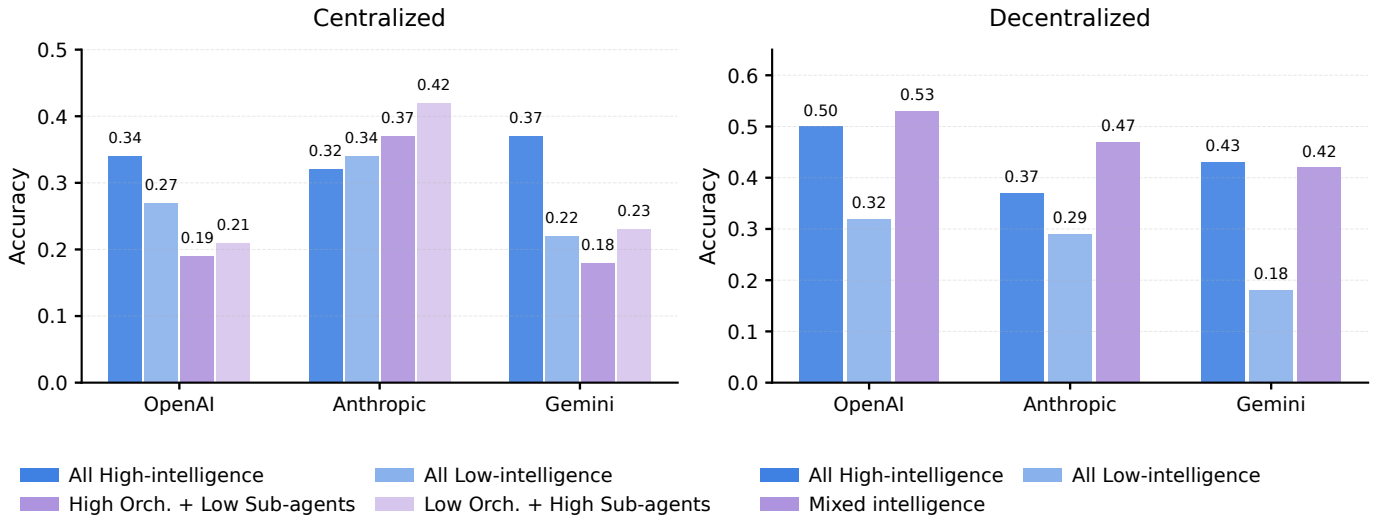
Coordination metrics across architectures and families (N=260 configurations across six benchmarks). Metric values are architecture-level constants measured from execution-trace analysis and applied uniformly across all benchmarks. All systems run under matched per-system compute ceilings. Trace-level error amplification (A_e) is measured from execution-trace token analysis.

https://doi.org/10.1038/s42256-026-01268-y



Extended Data Fig. 1 | Cost-performance trade-offs across model families and coordination architectures. Cost-performance trade-offs across model families and architectures. Data from BrowseComp-Plus, Finance Agent, PlanCraft, and WorkBench (SWE-bench Verified and Terminal-Bench use Docker-based evaluation with different cost structures). Comparative analysis of single-agent and multi-agent architectures (Independent, Decentralized, Centralized, and Hybrid) across three LLM families. Each point represents the mean agentic performance (%) versus normalized cost per experiment (USD), with horizontal and vertical error bars denoting standard error of the mean in cost and performance, respectively. The optimal coordination pattern differs across model families: OpenAI models show consistent gains from Centralized and Hybrid configurations despite higher costs; Google models display marginal improvements with a clear efficiency plateau; Anthropic models show higher variance and occasional MAS underperformance. Cross-family differences should be interpreted as empirical cost-performance signatures rather than isolated mechanisms.

https://doi.org/10.1038/s42256-026-01268-y



Extended Data Fig. 2 | Agent-heterogeneity effects on multi-agent performance on BrowseComp-Plus. Agent heterogeneity effects on multi-agent performance. Performance comparison of centralized (orchestrator-subagents) and decentralized (peer debate with voting) architectures on BrowseComp-Plus across three LLM families. Each bar shows mean accuracy over n=100 independent task instances per (architecture, configuration) cell; bars are single-run point estimates, so no error bars are shown. Bootstrap CIs for the separate SWE-bench Verified and Terminal-Bench validation analyses are reported in Supplementary Table S11. High-capability models: GPT-5, Claude Sonnet 4.5, Gemini-2.5 Pro; low-capability: GPT-5 nano, Claude Sonnet 3.7, Gemini-2.0 Flash. These exploratory results do not show that model mixing bypasses the capability-saturation threshold. Canonical Δ values relative to the homogeneous strong-model baseline are in Supplementary Table S7: (1) no heterogeneous centralized configuration outperforms the homogeneous strong-model baseline (mean $\Delta = -0.126$ over seven configs); (2) decentralized mixed teams approach or slightly exceed it (mean $\Delta = +0.020$ over six configs), largely from stronger constituent models; (3) high-capability sub-agents outperform high-capability orchestrators.