To aggregate or Not to Aggregate
Test-Time Aggregation Beyond Verifier-Friendly Benchmarks

Shreyas Singh **Manoj Guduru** **Pradeep Moturi** **Kunal Singh**
Fractal AI Research

Abstract

Aggregation-based test-time scaling has produced strong gains on competition mathematics and code generation, but it remains unclear whether those gains transfer beyond verifier-friendly benchmarks, under which task conditions aggregation helps, and when one-step aggregation is sufficient relative to recursive aggregation. We study these questions across structured reasoning, knowledge-intensive reasoning, and medical reasoning, spanning proof-style mathematics, expert-level STEM reasoning, social-science knowledge tasks, BrowseComp-style information seeking, and both tool-free and tool-integrated regimes. Aggregation is effective when sampled trajectories contain recoverably complementary information: complementary reasoning progress in structured reasoning, or complementary retrieved evidence in tool-integrated knowledge/evidence-seeking. structured reasoning and tool-integrated knowledge/evidence-seeking recover 48% and 57% of available headroom on average, whereas medical reasoning without tools recovers only 21% despite a comparable diversity window. Tool use improves medical base performance, but aggregation remains weak because trajectories more often reflect competing clinical interpretations than composable intermediate progress. Within favorable regimes, aggregation type also matters: RSA is most useful for open-ended proof generation, whereas SSA captures most of the gain in tool-integrated knowledge/evidence-seeking at lower cost. Aggregation value therefore depends jointly on task structure, tool access, and aggregation type rather than following a uniform test-time scaling law across domains.

1. Introduction

The current wave of agentic AI is increasingly defined by systems that operate through tools and interfaces, including coding agents, deep-research systems, and computer-use agents (Li et al., 2025a; Hong et al., 2023; Singh et al., 2025c; Yao et al., 2023; Zhou et al., 2023). Across these settings, a recurring systems pattern is to sample multiple candidate trajectories, tool-use traces, or search branches and then select or synthesize a final answer. Aggregation-based test-time scaling formalizes this pattern by allocating additional inference-time compute to multiple trajectories and then selecting or combining them (Snell et al., 2024; Wang et al., 2024; Li et al., 2025b; Venkatraman et al., 2025). This approach has produced strong gains on competition mathematics and code generation, where correctness is comparatively easy to verify and partially correct trajectories can often be identified or recombined (Zhang et al., 2025; Singh et al., 2026; Venkatraman et al., 2025). The central open question is whether the same mechanism transfers beyond these verifier-friendly settings.

This paper studies three related questions. First, does aggregation improve performance outside competition math and coding? Second, if it does, under which task conditions does it help? Third, when aggregation is useful, when is a single aggregation pass sufficient and when is recursive aggregation warranted? These questions are domain-dependent. Structured reasoning tasks may admit complementary partial reasoning across trajectories; tool-integrated knowledge-intensive tasks may benefit from complementary retrieved evidence; medical reasoning often presents a qualitatively different regime in which trajectories encode competing global interpretations rather than composable intermediate progress.

We answer these questions through a cross-domain study of aggregation-based test-time scaling over proof-style mathematics, expert-level STEM reasoning, social-science knowledge tasks, BrowseComp-style information seeking, and medical reasoning. We evaluate open-weight and frontier models in tool-free settings, and open-weight models in tool-integrated settings (Luong et al., 2025; Wei et al., 2025; Center for AI Safety and Scale AI and collaborators, 2025; Zuo et al., 2025; Wu et al., 2025). Our analysis uses two diagnostics: the diversity window, which measures the headroom between Pass@1 and Pass@8, and Aggregation Yield, which measures how much of that headroom an aggregation method recovers.

Our results are domain-specific rather than uniform. Struc-

tured reasoning and tool-integrated knowledge/evidence-seeking are the most favorable regimes because independently sampled trajectories often contribute complementary reasoning progress or complementary retrieved evidence. Medical reasoning is the clearest unfavorable regime: tool use substantially improves underlying performance, but aggregation gains remain weak because trajectory diversity is less complementary. Within favorable regimes, aggregation type also matters: SSA is the preferred default in tool-integrated knowledge/evidence-seeking, where a single evidence-aware pass captures most of the gain, whereas RSA is mainly warranted in open-ended proof generation.

Our contributions are:

1. We provide a broad empirical study of aggregation across structured reasoning, knowledge/evidence-seeking, and medical reasoning in tool-free and tool-integrated regimes.
2. We show that aggregation is effective when trajectory diversity is recoverably complementary: complementary reasoning progress in structured reasoning and complementary retrieved evidence in tool-integrated knowledge/evidence-seeking.
3. We identify medical reasoning as the clearest counterexample: substantial multi-sample headroom often fails to convert into aggregation gains because trajectories are less compositionally complementary, even with improved tool-enabled base performance.
4. We distinguish when SSA and RSA are warranted: SSA is the practical default in tool-integrated knowledge/evidence-seeking, whereas RSA is most useful in proof-style structured reasoning.

2. Related Work

Test-time scaling and parallel reasoning. Inference-time scaling spans both sequential depth along a single trajectory and parallel breadth across many trajectories. Recent work shows that search, reranking, and self-verification can substantially improve closed-ended math and coding performance (Snell et al., 2024; Wang et al., 2024; Pan et al., 2025; Lian et al., 2025). Related program-aided methods push on the sequential axis, for example by decomposing Olympiad-style math into executable subproblems (Singh et al., 2025b). Our focus is complementary: rather than proposing a new aggregation method, we ask when parallel aggregation is worthwhile beyond the benchmark regimes on which it is usually evaluated.

Verification and candidate selection. Aggregation depends on deciding which sampled trajectory to trust. Recent work explores verifier-guided search, self-verification, and pairwise comparison as mechanisms for improving candidate selection (Snell et al., 2024; Zhang et al., 2025; Singh et al., 2026). In code generation, stress-testing based filters provide another example of using executable feedback to narrow the gap between Pass@k and final selection (Singh et al., 2025a). These methods are especially effective in verifier-friendly domains; our study asks how well such selection-based aggregation transfers once trajectories differ not only in correctness but also in evidence, framing, or diagnostic interpretation.

Aggregation and synthesis. Beyond selecting a single candidate, recent work studies whether multiple trajectories can be synthesized into a better final answer. Generative self-aggregation and recursive self-aggregation are motivated by the possibility that different trajectories contain complementary partial reasoning (Li et al., 2025b; Venkatraman et al., 2025). Our study complements this literature by showing that such complementarity is highly domain-dependent.

Tool-augmented and agentic reasoning. In tool-enabled settings, rollouts differ not only in reasoning quality but also in the evidence they retrieve. Recent work spans ReAct-style reasoning–action loops (Yao et al., 2023; Zhou et al., 2023), deep-research systems (Li et al., 2025a; Singh et al., 2025d), and GUI/computer-use agents (Hong et al., 2023; Singh et al., 2025c; Wei et al., 2025). These settings make aggregation qualitatively different because trajectories can differ in retrieved evidence, action traces, and interface state rather than only in internal reasoning. We study this regime directly by comparing tool-free and tool-integrated aggregation across reasoning, knowledge-seeking, and medical tasks.

3. Experimental Setup

We study aggregation along three axes: *domain*, *model family*, and *tool regime*. This design lets us separate whether aggregation helps at all from which aggregation strategy is warranted in each setting.

3.1. Models and Task Groups

We evaluate open-weight Qwen-3-8B, Qwen-3-30B, Qwen-3.5-35B, GPT-OSS-20B, and GPT-OSS-120B, together with frontier GPT-5, Claude 4.6 Opus, and Gemini-3-Pro. Tool-free experiments use the full set; tool-integrated experiments focus on GPT-OSS-20B, Qwen-3.5-35B, and GPT-OSS-120B. All frontier-model runs use medium reasoning effort. For compact figure labels, we refer to GPT-OSS-20B, Qwen-3.5-35B, GPT-OSS-120B, Gemini-3-Pro, GPT-5, and Claude 4.6 Opus as O20, Q35, O120, G3, G5, and C46, respectively.

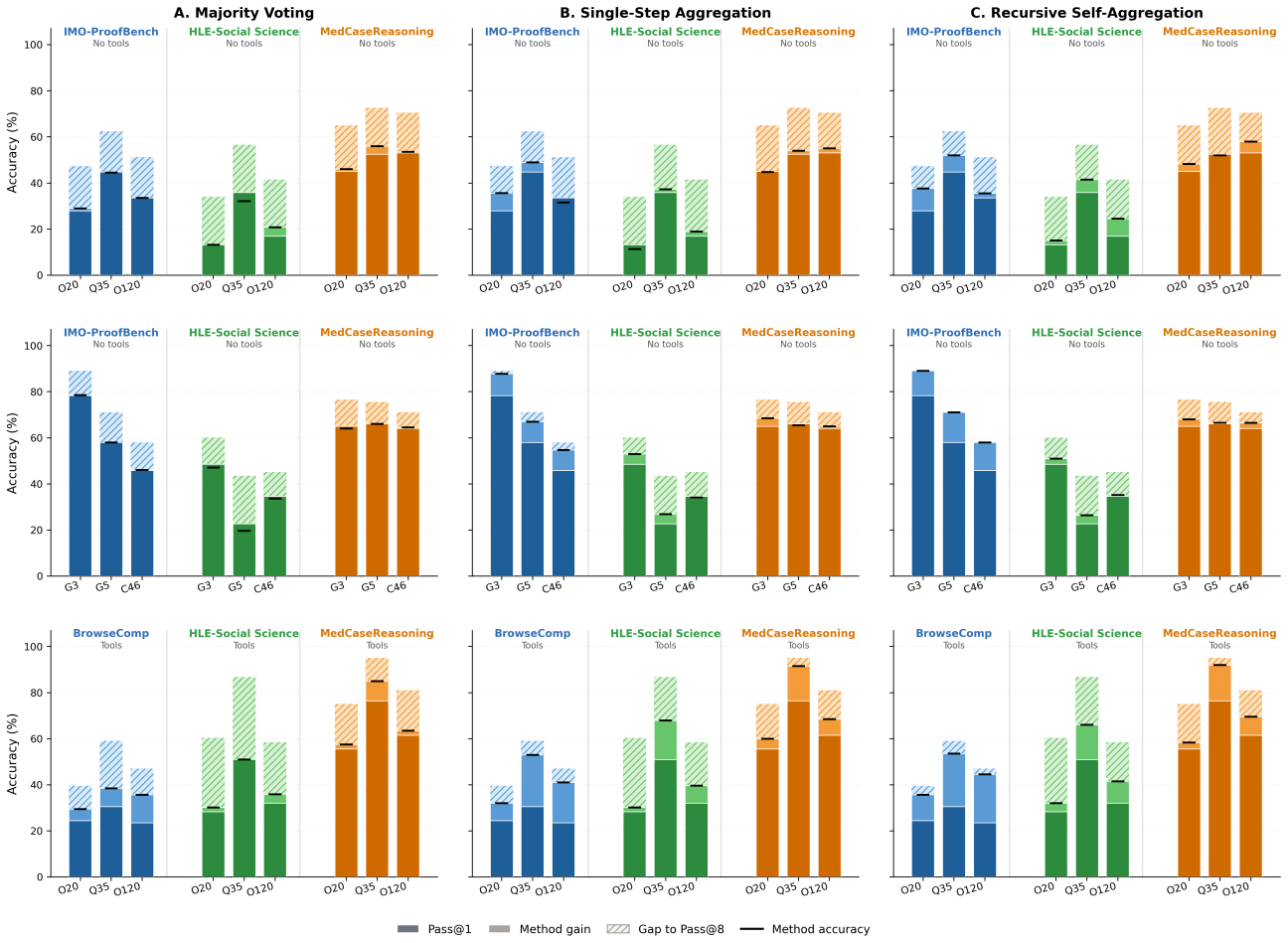


Figure 1. Representative benchmark-level comparison across open-weight no-tool (top), frontier no-tool (middle), and open-weight tool-enabled (bottom) settings. Model proxies are O20 = GPT-OSS-20B, Q35 = Qwen-3.5-35B, O120 = GPT-OSS-120B, G3 = Gemini-3-Pro, G5 = GPT-5, and C46 = Claude 4.6 Opus. ProofBench is the clearest RSA-favorable case, tool-integrated knowledge/evidence-seeking favors SSA, and medical reasoning improves with tools but remains weakly recoverable under aggregation.

We group tasks into three regimes. **Structured reasoning** includes IMO-ProofBench and HLE-STEM, plus tool-augmented HLE-STEM. **Medical reasoning** includes MedXpertQA and MedCaseReasoning. **Knowledge and information-seeking** includes HLE-Social Science without tools and, with tools, HLE-Social Science together with BrowseComp (Luong et al., 2025; Center for AI Safety and Scale AI and collaborators, 2025; Zuo et al., 2025; Wu et al., 2025; Wei et al., 2025). In the analysis, tool-augmented HLE-Social Science and BrowseComp are treated jointly as a knowledge/evidence-seeking regime, whereas tool-augmented HLE-STEM remains grouped with structured reasoning.

3.2. Problem Setup

Given an input x and a base model p_θ, we sample $k = 8$ independent worker trajectories

$$\mathcal{T}(x) = \{T_1, \ldots, T_k\}.$$

Each trajectory produces a final answer y_i. In the most general tool-integrated setting,

$$T_i = (x, r_{i,1}, a_{i,1}, o_{i,1}, \ldots, r_{i,m_i}, a_{i,m_i}, o_{i,m_i}, y_i),$$

where $r_{i,j}$ denotes intermediate reasoning, $a_{i,j}$ a tool or environment action, and $o_{i,j}$ the resulting observation. In tool-free settings, the action–observation terms are absent.

The aggregation problem is to map the set of worker trajectories to a single prediction

$$f_\phi(x, \mathcal{T}(x)) \mapsto \hat{y}.$$

We evaluate tool-free and tool-integrated aggregation separately, since in the latter case trajectories differ not only in reasoning but also in the external evidence they uncover.

3.3. Aggregation Methods

We compare four aggregation methods over the sampled worker set $\mathcal{T}(x) = \{T_1, \ldots, T_k\}$.

Majority Voting (MV). As a simple baseline, we select the most frequent final answer among the sampled workers:

$$\hat{y}_{\mathrm{MV}} = \arg\max_{y} \sum_{i=1}^{k} \mathbf{1}[y_i = y].$$

MV uses only the final answers and ignores the reasoning traces.

Single-Step Aggregation (SSA). SSA is a single LLM call over all sampled worker trajectories:

$$\hat{y}_{\mathrm{SSA}} = A_\phi(x, \mathcal{T}(x)).$$

SSA is our main one-step aggregation baseline.

Recursive Self-Aggregation (RSA). RSA applies SSA recursively over randomly sampled subsets. Let K denote the subset size, N the number of subsets sampled at each round, and P the number of aggregation rounds. Starting from the initial pool

$$\mathcal{P}^{(0)} = \mathcal{T}(x),$$

each round samples N subsets

$$S_1^{(p)}, \ldots, S_N^{(p)} \subseteq \mathcal{P}^{(p)}, \qquad |S_j^{(p)}| = K,$$

applies SSA independently to obtain

$$\tilde{y}_j^{(p+1)} = A_\phi(x, S_j^{(p)}), \qquad j = 1, \ldots, N,$$

and forms the next pool

$$\mathcal{P}^{(p+1)} = \{\tilde{y}_1^{(p+1)}, \ldots, \tilde{y}_N^{(p+1)}\}.$$

After P rounds, we apply one final SSA call to the last pool to obtain

$$\hat{y}_{\mathrm{RSA}} = A_\phi(x, \mathcal{P}^{(P)}).$$

RSA is intended to improve answers by repeatedly aggregating smaller subsets of candidate trajectories (Venkatraman et al., 2025).

Compressed Evidence Aggregation (CEA). In tool-integrated settings, full trajectories are often too long to aggregate directly. Our main tool-based method is **Compressed Evidence Aggregation (CEA)**: for each worker T_i with final answer y_i, we select only the answer-relevant tool evidence and the final reasoning step, constructing

$$e_i = E_\phi(x, T_i, y_i),$$

where e_i contains the selected evidence and r_i^{final} denotes the final reasoning step immediately preceding y_i. Aggregation is then performed over the compressed worker representations:

$$\hat{y}_{\mathrm{CEA}} = A_\phi\big(x, \{(e_i, r_i^{\mathrm{final}}, y_i)\}_{i=1}^{k}\big).$$

We use CEA in all main tool-integrated experiments.

3.4. Evaluation Protocol

For each model–task pair, we sample $k = 8$ independent worker trajectories using temperature-based decoding. For the open-weight reasoning models, we use the family-recommended temperatures: 0.7 for GPT-OSS models and 1.5 for Qwen reasoning models. We report Pass@1, Pass@8, and post-aggregation accuracy for each method. Pass@8 measures whether at least one of the k workers is correct and therefore serves as an oracle upper bound on recoverable performance from the sampled worker pool. For IMO-ProofBench, Pass@8 is computed from per-rollout proof-quality judgments, and the MV column is a mode-based proxy over rollout proof scores. All benchmarks except IMO-ProofBench are judged with an LLM exact-match-style scorer; IMO-ProofBench uses rubric-based proof evaluation. We also track the total token cost of each method, including worker generation and all aggregation calls; for CEA this includes both evidence compression and the final aggregation call.

4. Analysis

We analyze aggregation performance using two quantities: the diversity window, defined as Pass@8 − Pass@1, and Aggregation Yield (AY), which measures the fraction of this headroom recovered by a given aggregation method. The central distinction is between raw multi-sample headroom and whether the resulting trajectory variation is recoverably complementary. These quantities are used throughout the section to compare aggregation behavior across domains, model families, and tool regimes.

4.1. Diversity Window and Aggregation Yield

For each model–task pair, we sample $k = 8$ worker trajectories and report Pass@1, Pass@8, and post-aggregation accuracy. Pass@1 measures the expected accuracy of a single sampled worker. Pass@8 measures whether at least one

of the k sampled workers is correct. The difference

$$\text{Pass@8} - \text{Pass@1}$$

defines the **Diversity window (DW)**. It measures the amount of recoverable headroom available in the sampled worker pool.

Raw aggregation gains are not directly comparable across tasks with different amounts of available headroom. We therefore define the **Aggregation Yield (AY)** of a method $\mathcal{S}$ as

$$\mathrm{AY}(\mathcal{S}) = \frac{\mathrm{Acc}(\mathcal{S}) - \text{Pass@1}}{\text{Pass@8} - \text{Pass@1}},$$

where $\mathrm{Acc}(\mathcal{S})$ is the final accuracy after applying aggregation method $\mathcal{S}$. AY is equal to 0 when aggregation provides no gain over a single sample, equal to 1 when it fully recovers the available headroom, and negative when aggregation degrades performance. For readability, we report AY as the percentage of recoverable headroom captured by the best aggregation method. AY therefore distinguishes cases in which aggregation fails because little headroom exists from cases in which substantial headroom exists but is not exploitable.

4.2. Aggregation Depends on Complementarity, Not Headroom Alone

The main empirical distinction is not the size of the diversity window by itself, but whether the induced trajectory diversity is complementary and recoverable. Table 1 summarizes the regime-level pattern. Structured reasoning without tools has average DW of 16.44 pp and average AY of 48%. Tool-augmented HLE-STEM, treated as a structured variant, has average DW of 33.33 pp and average AY of 42%. The strongest recoverability appears in tool-integrated knowledge/evidence-seeking, which has average DW of 26.89 pp and average AY of 57%. By contrast, knowledge-intensive reasoning without tools and medical reasoning without tools each recover only 21% of available headroom despite non-trivial diversity windows; medical reasoning with tools improves to 37%, but remains well below the knowledge/evidence- seeking tool regime.

Table 1. Regime summary. DW = Pass@8 − Pass@1; AY = recovered headroom.

Regime	DW	AY (%)
Structured (no tools)	16.44	48
Structured (tools)	33.33	42
Knowledge (no tools)	19.51	21
Knowledge (tools)	26.89	57
Medical (no tools)	15.77	21
Medical (tools)	22.25	37

These contrasts are consistent with distinct forms of trajectory variation. In structured reasoning tasks, sampled trajectories often contain different partial advances toward the same solution. In no-tool knowledge-intensive settings, performance is often bottlenecked by missing or weakly represented facts, limiting what aggregation can recover. In tool-integrated knowledge-intensive settings, trajectories differ in the evidence they retrieve, so aggregation benefits from selecting and consolidating the most informative evidence-bearing rollouts. In medical reasoning, by contrast, trajectories more often represent competing global interpretations, making disagreement less amenable to aggregation.

The main methodological implication is that raw headroom is not sufficient to predict aggregation gains. Benchmarks with similar Pass@8–Pass@1 gaps can have substantially different post-aggregation outcomes because the underlying trajectory diversity is structurally different.

4.3. Structured Reasoning: Complementary Reasoning Progress Favors RSA

Structured reasoning is the most favorable no-tool regime because sampled trajectories often contain compatible partial derivations, lemmas, or decompositions that can be identified or recombined. Across the structured no-tool rows in Table 1, average AY is 48%. On HLE-STEM, aggregation yields large improvements for frontier models: GPT-5 improves from 38.51% to 50.00%, and Claude 4.6 Opus improves from 42.65% to 58.10%.

IMO-ProofBench is more favorable still to recursive aggregation. RSA is the best method in every ProofBench row in Table 2, including improvements from 27.97% to 37.67% for GPT-OSS-20B, from 58.01% to 71.00% for GPT-5, and from 45.79% to 58.00% for Claude 4.6 Opus. The contrast with HLE-STEM is informative: on HLE-STEM, SSA captures most of the available gain and exceeds RSA for Gemini-3-Pro and GPT-5, whereas on IMO-ProofBench iterative aggregation is consistently superior. This pattern is consistent with proof generation placing greater value on combining reasoning progress distributed across multiple trajectories.

Tool access does not alter the basic character of HLE-STEM. Across the three tool-augmented HLE-STEM rows in Table 5, average DW is 33.33 pp and average AY is 42%. Both SSA and RSA improve substantially over Pass@1, but RSA adds only 0.13 pp over SSA on average. Thus, even in tool-augmented structured reasoning, most recoverable gain is captured by a single aggregation pass; recursion is most clearly justified in the proof-generation setting.

Tools improve medical reasoning, but they do not change its qualitative position. Across the six open-weight medical settings with tools, average AY rises from 21% to 37%, still well below the 57% observed for the knowledge-intensive

Table 2. **No-tool results on HLE-STEM, HLE-Social Science, and IMO-ProofBench.** Best aggregation result per row is shown in **bold**; values are accuracy (%).

	HLE-STEM					HLE-Social Science					IMO-ProofBench				
Model	P@1	P@8	MV	SSA	RSA	P@1	P@8	MV	SSA	RSA	P@1	P@8	MV	SSA	RSA
Qwen-3-8B	1.35	12.16	1.35	2.9	**3.20**	3.77	30.19	11.32	**13.21**	**13.21**	19.67	34.00	20.75	23.67	**25.00**
GPT-OSS-20B	12.16	32.43	9.46	13.51	**18.92**	13.21	33.96	13.21	11.32	**15.09**	27.97	47.33	28.92	35.67	**37.67**
Qwen-3-30B	8.11	22.97	6.76	8.22	**12.33**	11.32	32.08	9.43	9.43	**11.32**	27.67	38.33	25.67	29.48	**32.00**
Qwen-3.5-35B	36.49	55.41	36.49	35.71	**36.99**	35.85	56.60	32.08	37.25	**41.51**	44.80	62.40	44.50	49.00	**52.00**
GPT-OSS-120B	20.11	40.54	15.28	21.62	**24.32**	16.98	41.51	20.75	18.87	**24.53**	33.50	51.17	33.53	31.50	**35.50**
Gemini-3-Pro	61.66	82.43	64.86	**71.00**	68.00	48.45	60.10	47.15	**53.00**	51.00	78.27	89.00	78.50	87.80	**89.00**
GPT-5	38.51	60.81	33.78	**50.00**	47.14	22.73	43.52	19.69	**26.94**	26.42	58.01	71.00	58.00	67.00	**71.00**
Claude 4.6 Opus	42.65	61.76	42.60	54.00	**58.10**	34.65	45.08	33.68	34.19	**35.23**	45.79	58.00	46.00	54.67	**58.00**

Table 3. **No-tool medical reasoning.** Best result in **bold**; accuracy (%).

	MedXpertQA				
Model	**P@1**	**P@8**	**MV**	**SSA**	**RSA**
Qwen-3-8B	16.50	33.50	**20.50**	18.50	20.00
GPT-OSS-20B	30.50	56.00	34.00	**35.50**	33.50
Qwen-3-30B	26.00	40.00	**27.50**	23.50	24.50
Qwen-3.5-35B	40.50	60.00	**47.50**	46.00	41.21
GPT-OSS-120B	28.50	57.50	32.50	38.50	**39.50**
Gemini-3-Pro	71.00	78.00	70.00	71.00	**73.00**
GPT-5	57.00	66.00	55.00	56.00	**56.50**
Claude 4.6 Opus	51.75	59.00	51.00	**52.00**	50.50
	MedCaseReasoning				
Qwen-3-8B	30.50	49.50	35.00	**35.86**	35.68
GPT-OSS-20B	45.00	65.00	46.00	44.72	**48.24**
Qwen-3-30B	36.50	56.00	38.50	37.50	**40.00**
Qwen-3.5-35B	52.50	72.50	**56.00**	54.00	52.00
GPT-OSS-120B	53.00	70.50	53.50	55.00	**58.00**
Gemini-3-Pro	65.00	76.50	64.00	**68.50**	68.00
GPT-5	66.00	75.50	66.00	65.50	**66.50**
Claude 4.6 Opus	64.00	71.00	64.50	65.00	**66.50**

Table 4. **Tool-integrated medical reasoning.** CEA aggregation; best result in **bold**; accuracy (%).

	MedXpertQA				
Model	**P@1**	**P@8**	**MV**	**SSA**	**RSA**
GPT-OSS-20B	38.50	67.50	**43.00**	39.50	41.00
Qwen-3.5-35B	41.50	66.00	**49.00**	48.00	48.00
GPT-OSS-120B	44.00	66.50	47.00	49.00	**50.00**
	MedCaseReasoning				
GPT-OSS-20B	55.50	75.00	57.50	**60.00**	58.38
Qwen-3.5-35B	76.50	95.00	85.00	91.50	**92.00**
GPT-OSS-120B	61.50	81.00	63.50	68.50	**69.50**

tool regime. Table 4 shows large gains on MedCaseReasoning, especially for Qwen-3.5-35B and GPT-OSS-120B, but medical trajectories remain markedly less aggregation-friendly than tool-integrated knowledge/evidence-seeking.

4.4. To aggregate or not to aggregate?

Different favorable regimes do not call for the same aggregation strategy. Signed RSA-minus-SSA deltas are largest and most consistent in structured reasoning without tools, driven primarily by ProofBench. This is the clearest setting in which recursive aggregation recovers complementary reasoning progress that a single aggregation pass leaves unused. Figure 2 summarizes the contrast across regimes.

4.5. Knowledge and Evidence-Seeking: Complementary Evidence Favors SSA

Knowledge-intensive reasoning splits cleanly into two regimes. Without tools, HLE-Social Science is a boundary condition for aggregation: average DW is 19.51 pp, but average AY is only 21% (Table 1). Gains exist, but they remain modest relative to available headroom. For example, GPT-5 improves from 22.73% to 26.94%, and Claude 4.6 Opus improves only from 34.65% to 35.23%. The limiting factor is factual coverage: when the relevant fact is absent or only weakly represented, repeated sampling does not create the kind of complementary structure that aggregation can reliably exploit.

The picture changes once evidence acquisition is enabled. Treating tool-augmented HLE-Social Science together with BrowseComp as a single knowledge/evidence-seeking tool regime yields average DW of 26.89 pp and average AY of 57%, the highest value in Table 1. BrowseComp provides the clearest illustration: the best aggregation result improves over Pass@1 by 11.18 pp for GPT-OSS-20B, 23.00 pp for Qwen-3.5-35B, and 21.00 pp for GPT-OSS-120B. Tool-augmented HLE-Social Science shows the same qualitative shift, especially for Qwen-3.5-35B, which improves from 50.94% to 67.92%. The underlying mechanism is complementary retrieved evidence: different trajectories surface different relevant snippets, and aggregation succeeds by consolidating them.

Across these six knowledge-intensive tool rows, SSA improves over Pass@1 by 12.32 pp on average, while RSA

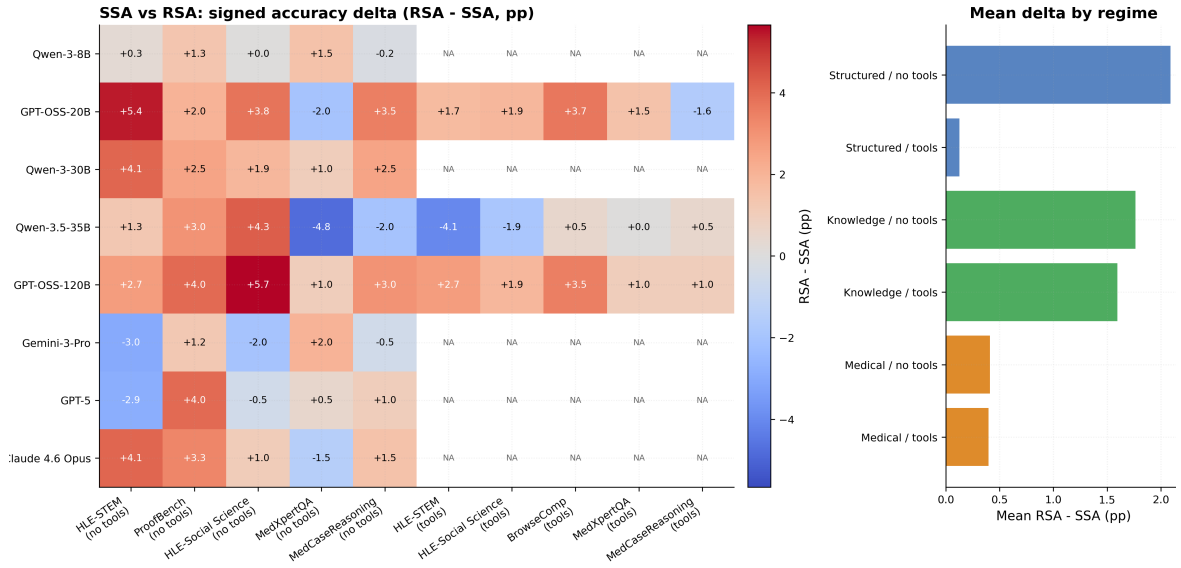


Figure 2. **Signed RSA-minus-SSA deltas across regimes.** Positive values favor RSA. Recursive aggregation is most consistently helpful in structured reasoning without tools, while tool-integrated knowledge/evidence-seeking and medical reasoning remain close to zero, favoring SSA as the default operating point.

improves by 13.91 pp, only 1.60 pp more. Most of the available gain therefore arises from a single evidence-aware aggregation step rather than from deeper recursive synthesis. Accordingly, SSA is the preferred operating point in this regime: RSA can add small increments, but the principal bottleneck is evidence discovery and retention rather than the recombination of partially correct internal reasoning.

4.6. Medical Reasoning: Tools Help, Aggregation Remains Weak

Medical reasoning is the clearest counterexample. These benchmarks often exhibit substantial headroom, but much less of it is recoverable by aggregation. Across the no-tool medical rows, average DW is 15.77 pp, close to the structured-reasoning average, but average AY is only 21% (Table 1).

On MedXpertQA, Qwen-3.5-35B has a 19.50 pp diversity window (40.50% to 60.00%) but reaches only 47.50% after aggregation, and GPT-5 remains below Pass@1 after aggregation despite Pass@8 of 66.00%. Majority voting is also unusually competitive on MedXpertQA. This pattern is consistent with medical trajectories differing more often in global diagnostic framing than in separable intermediate steps.

Outside structured reasoning, the deltas are small. In tool-integrated knowledge/evidence-seeking, RSA adds little once SSA has already surfaced the relevant evidence. In medical reasoning, deltas are smallest in both tool-free and tool-integrated settings, consistent with the broader finding that medical trajectories are weakly complementary. Figure 2 therefore identifies the main regime in which recursion changes the outcome rather than merely adding cost.

4.7. Accuracy–Cost Tradeoffs

Token accounting clarifies the same allocation question. Because SSA aggregates all eight workers, its total cost is nearly an order of magnitude above Pass@1 in our open-weight runs: about $9.8\times$ a single rollout on average without tools and $9.3\times$ with tools. RSA then adds a further $1.6\times$ over SSA on average without tools and $1.2\times$ with tools. The relevant question is therefore not whether recursive aggregation can ever help, but in which regimes its additional cost changes the accuracy–cost frontier in a meaningful way. Figure 3 shows this across representative open-weight models and datasets.

Three frontier patterns are consistent. First, when success depends on retrieved evidence, tool use shifts the frontier more than deeper aggregation. In HLE-Social Science and BrowseComp, the tool-enabled curves in Figure 3 sit well above their no-tool counterparts for the same models, indicating that the dominant gain comes from better evidence discovery rather than from deeper no-tool synthesis.

Second, SSA is the preferred operating point once complementary evidence has been surfaced. Across HLE-Social Science, BrowseComp, and tool-enabled MedCaseReasoning, RSA adds little or nothing beyond SSA.

Third, RSA is justified primarily in proof-style structured reasoning. On ProofBench in Table 2, Qwen-3.5-35B improves from 49.00% under SSA to 52.00% under RSA, and GPT-OSS-120B improves from 31.50% to 35.50%. This is

Table 5. **Tool-integrated results on HLE-STEM, HLE-Social Science, and BrowseComp.** Tool-based aggregation uses CEA by default. Best aggregation result per row is shown in **bold**; values are accuracy (%).

	HLE-STEM					HLE-Social Science					BrowseComp				
Model	P@1	P@8	MV	SSA	RSA	P@1	P@8	MV	SSA	RSA	P@1	P@8	MV	SSA	RSA
GPT-OSS-20B	17.57	51.35	25.68	27.03	**28.77**	28.30	60.38	30.19	30.19	**32.08**	24.50	39.50	29.50	32.00	**35.68**
Qwen-3.5-35B	29.73	66.22	39.19	**47.30**	43.24	50.94	86.79	50.94	**67.92**	66.04	30.50	59.00	38.50	53.00	**53.50**
GPT-OSS-120B	24.32	54.05	31.94	35.14	**37.84**	32.08	58.49	35.85	39.62	**41.51**	23.50	47.00	35.68	41.00	**44.50**

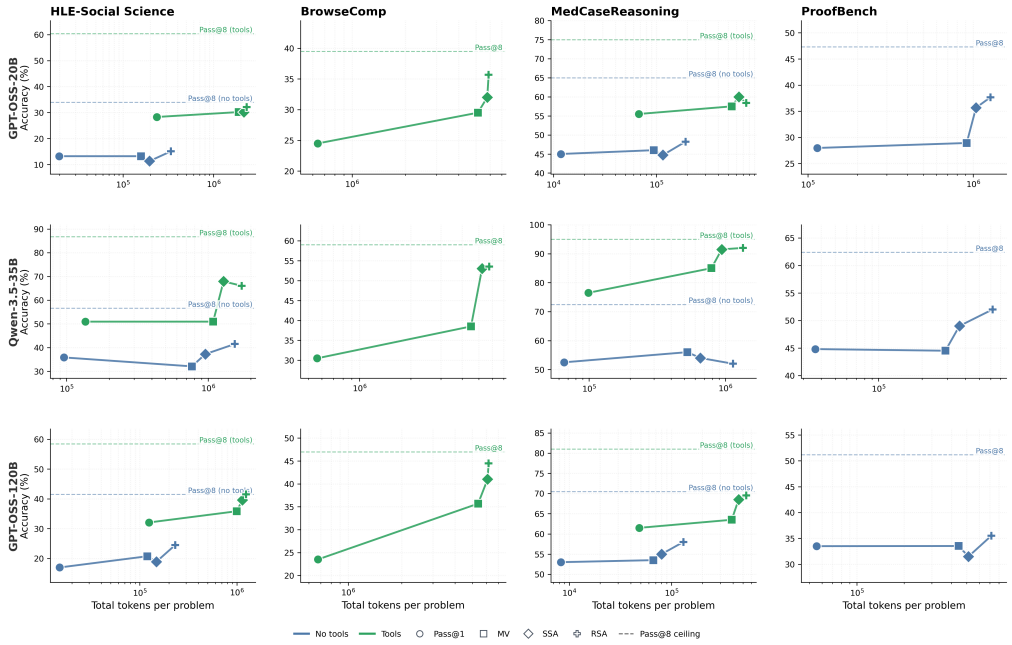


Figure 3. Accuracy–cost Pareto curves for the open-weight models. RSA most clearly extends the frontier on ProofBench; in HLE-Social Science, BrowseComp, and MedCaseReasoning, tool use dominates and SSA captures most practical gain.

the clearest regime in which recursion materially extends the frontier beyond SSA.

Operationally, use tools first when distinct evidence can be retrieved, default to SSA once such evidence is available, and reserve RSA for tool-free structured reasoning.

5. Conclusion

Aggregation transfers beyond verifier-friendly benchmarks only when sampled trajectories are complementary: structured reasoning benefits from complementary reasoning progress, knowledge/evidence-seeking from complementary retrieved evidence, and medical reasoning remains unfavorable even with tools. Practically, use tools first when evidence matters, prefer SSA for knowledge/evidence-seeking, and reserve RSA for tool-free structured reasoning, especially open-ended proof generation.

References

Center for AI Safety and Scale AI and collaborators. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Zhang, Y., Li, J., Xu, B., Dong, Y., Ding, M., and Tang, J. Cogagent: A visual language model for gui agents, 2023. URL https://arxiv.org/abs/2312.08914.

Li, B., Zhang, D., Wu, J., Yin, W., Tao, Z., Zhao, Y., Zhang, L., Shen, H., Fang, R., Xie, P., Zhou, J., and Jiang, Y. Parallelmuse: Agentic parallel thinking for deep information seeking. *arXiv preprint arXiv:2510.24698*, 2025a.

Li, Z., Feng, X., Cai, Y., Zhang, Z., Liu, T., Liang, C., Chen, W., Wang, H., and Zhao, T. Llms can generate a better answer by aggregating their own responses. *arXiv preprint arXiv:2503.04104*, 2025b.

Lian, L., Wang, S., Juefei-Xu, F., Fu, T.-J., Li, X., Yala, A., Darrell, T., Suhr, A., Tian, Y., and Lin, X. V. Threadweaver: Adaptive threading for efficient parallel reasoning in language models. *arXiv preprint arXiv:2512.07843*, 2025.

Luong, T., Hwang, D., Nguyen, H. H., Ghiasi, G., Chervonyi, Y., Seo, I., Kim, J., Bingham, G., Lee, J., Mishra, S., Zhai, A., Hu, C. H., Michalewski, H., Kim, J., Ahn, J., Bae, J., Song, X., Trinh, T. H., Le, Q. V., and Jung, J. Towards robust mathematical reasoning. *arXiv preprint arXiv:2511.01846*, 2025.

Pan, Y. et al. Learning adaptive parallel reasoning with language models. *arXiv preprint arXiv:2504.15466*, 2025.

Singh, H. et al. V1: Unifying generation and self-verification for parallel reasoners. https://harmandotpy.github.io/v1-verification/, 2026. Project page.

Singh, K., Bhowmick, S., Moturi, P., and Gollapalli, S. K. No stress no gain: Stress testing based self-consistency for olympiad programming. Published at the ICLR 2025 VerifAI: AI Verification in the Wild, 2025a. URL https://openreview.net/pdf?id=7SlCSjhBsq.

Singh, K., Biswas, A., Bhowmick, S., Moturi, P., and Gollapalli, S. K. Sbsc: Step-by-step coding for improving mathematical olympiad performance, 2025b. URL https://arxiv.org/abs/2502.16666.

Singh, K., Singh, S., and Khanna, M. Trishul: Towards region identification and screen hierarchy understanding for large vlm based gui agents, 2025c. URL https://arxiv.org/abs/2502.08226.

Singh, S., Singh, K., and Moturi, P. Fathom-deepresearch: Unlocking long horizon information retrieval and synthesis for slms, 2025d. URL https://arxiv.org/abs/2509.24107.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Venkatraman, S. et al. Recursive self-aggregation unlocks deep thinking in large language models. *arXiv preprint arXiv:2509.26626*, 2025.

Wang, J., Wang, J., Athiwaratkun, B., Zhang, C., and Zou, J. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*, 2024.

Wei, J., Sun, Z., Papay, S., McKinney, S., Han, J., Fulford, I., Chung, H. W., Passos, A. T., Fedus, W., and Glaese, A. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Wu, K., Wu, E., Thapa, R., Wei, K., Zhang, A., Suresh, A., Tao, J. J., Sun, M. W., Lozano, A., and Zou, J. Medcasereasoning: Evaluating and learning diagnostic reasoning from clinical case reports. *arXiv preprint arXiv:2505.11733*, 2025.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Zhang, F., Xu, J., Wang, C., Cui, C., Liu, Y., and An, B. Incentivizing llms to self-verify their answers. *arXiv preprint arXiv:2506.01369*, 2025.

Zhou, A., Yan, K., Shlapentokh-Rothman, M., Wang, H., and Wang, Y.-X. Language agent tree search unifies reasoning acting and planning in language models, 2023. URL https://arxiv.org/abs/2310.04406.

Zuo, Y. et al. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. *arXiv preprint arXiv:2501.18362*, 2025.

A. Qualitative Examples

We present three representative case studies. Each block includes an abridged question, representative trajectories, the aggregation outcome, and the interpretation that connects the example back to the main argument.

Case A. Structured reasoning: complementary proof progress is recoverable **ProofBench `q000011`, GPT-OSS-20B**

Question (abridged). Choose 1000 integers from $\{1, \ldots, 2000\}$ so that no chosen number divides another chosen number, and determine the smallest possible minimum element.

Raw pool. The eight sampled proofs are weak overall: 7 are judged wrong, 1 is judged partial, and the average normalized proof score is 0.0875. No single rollout contains a complete correct proof.

Representative trajectories.

- **p0001 (partial lower-bound argument).** Argues that if $m_A \leq 666$, then at least two numbers in $(1000, 2000]$ are divisible by m_A, and therefore concludes $m_A \geq 667$.
- **p0004 (incorrect constructive argument).** Introduces a chain-based decomposition of the divisibility poset, but the construction is flawed and the final answer is $m_A = 500$.
- **SSA.** Remains weak in this case: it effectively follows trajectory `p0001`, and the resulting proof is still judged wrong with normalized score 0.1.
- **RSA.** Synthesizes the lower-bound idea with an explicit construction excluding the critical multiple 1334, producing an almost-correct proof with normalized score 0.7.

Interpretation. This is the clearest qualitative case in which the useful information is distributed across trajectories. One proof contains the right lower bound, another contains part of the constructive idea, and recursive aggregation is able to combine them into a much stronger proof. This is precisely the regime in which RSA is warranted.

Case B. Tool-integrated knowledge/evidence-seeking: one evidence-bearing rollout is enough **BrowseComp `q000036`, Qwen-3.5-35B**

Question (abridged). Connect a November 2019 article about scoring methods in one sport to a January 2020 interview article about the same athlete, then identify that athlete's hometown.

Raw pool. Only one of the eight tool-using trajectories is correct, so the candidate accuracy is 0.125. The useful evidence is extremely sparse: the correct answer is supported by a single evidence-bearing path, while several other rollouts cluster around plausible but unsupported athletes.

Representative trajectories.

- **p0001 (wrong athlete cluster).** Interprets the query through a Patrick Mahomes path and answers *Tyler, Texas*.
- **p0002 (evidence-bearing trajectory).** Retrieves the relevant article pair and answers *Mallorca, Spain*. Its tool summary identifies a single decisive evidence item supporting that answer.
- **p0003 / p0006 (another wrong cluster).** Converge on Shaun White and answer *Carlsbad, California*.
- **SSA.** Selects the evidence-bearing trajectory and outputs *Mallorca, Spain*. The key point is that once the relevant evidence is present in the pool, a single aggregation step is sufficient.

Interpretation. This example shows why SSA is the preferred operating point in tool-integrated knowledge/evidence-seeking. Once a single rollout has retrieved the right evidence, the main challenge is simply to surface it. One evidence-bearing trajectory can already be enough.

Case C. Medical reasoning: disagreement is not complementary **MedCaseReasoning `q000037`, Qwen-3.5-35B**

Question (abridged). A 58-year-old woman presents with chronic necrotizing lymphadenitis, nasal chondritis, conjunctivitis, vasculitic skin lesions, persistent fevers, and positive *Bartonella henselae* serology. Identify the most likely diagnosis.

Raw pool. The candidate pool is not sparse: 4 of the 8 trajectories are correct, so the candidate accuracy is 0.5. The issue is not absence of correct trajectories, but disagreement between two coherent global interpretations of the case.

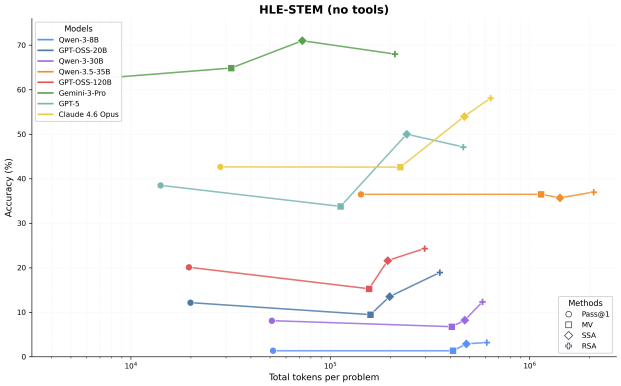
Representative trajectories.

- **p0001 (infectious framing).** Leans on the chronic lymphadenitis and serology and answers *Bartonella henselae infection (cat-scratch disease)*.
- **p0002 (autoimmune framing).** Treats nasal chondritis as pathognomonic and commits to *Relapsing Polychondritis*.
- **p0004 / p0006.** Again identify *Bartonella henselae infection*, showing that correct rollouts are present multiple times in the raw pool.
- **SSA.** Outputs *Relapsing Polychondritis*, following the wrong autoimmune interpretation.
- **RSA.** Also outputs *Relapsing Polychondritis*; recursion does not rescue the error.

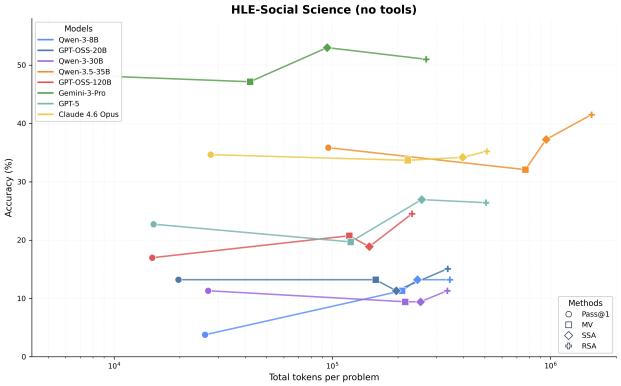
Interpretation. This is the clearest qualitative counterexample in the paper. The raw pool already contains several correct diagnoses, but the disagreement is not decomposable into complementary intermediate pieces. Instead, aggregation amplifies the wrong global interpretation, which is exactly why medical reasoning remains less aggregation-friendly than structured reasoning or tool-integrated evidence-seeking.

B. Full Pareto Frontiers

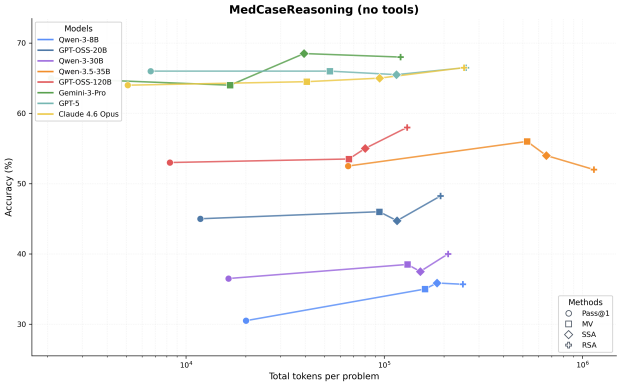
We include the full per-dataset Pareto frontiers used to support the accuracy–cost analysis in the main paper. Each panel plots Pass@1, MV, SSA, and RSA against total tokens per problem; dashed horizontal lines indicate the Pass@8 ceiling for the corresponding condition.



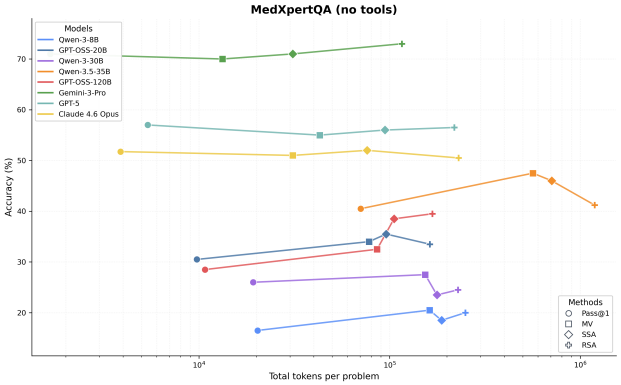
(a) HLE-STEM (no tools)



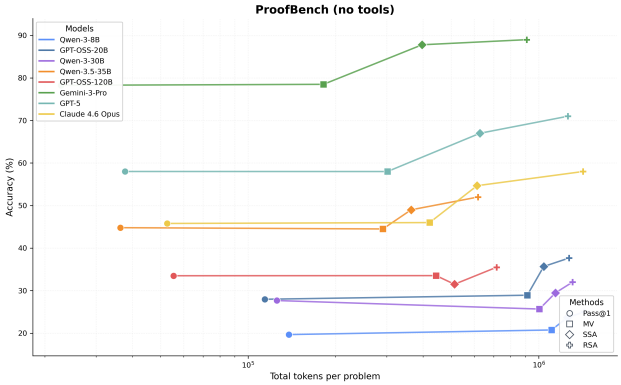
(b) HLE-Social Science (no tools)



(c) MedCaseReasoning (no tools)

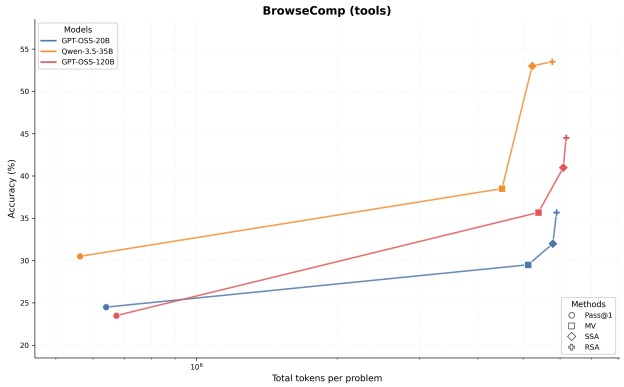


(d) MedXpertQA (no tools)

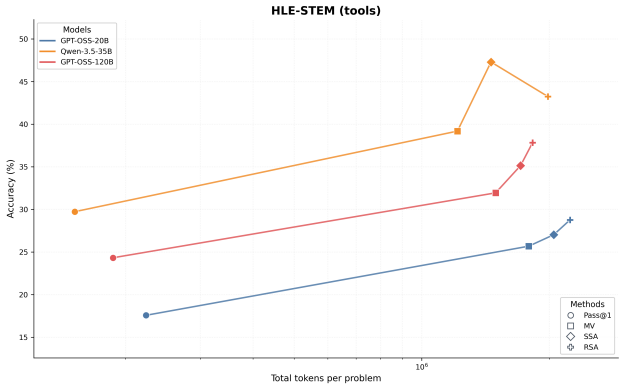


(e) ProofBench (no tools)

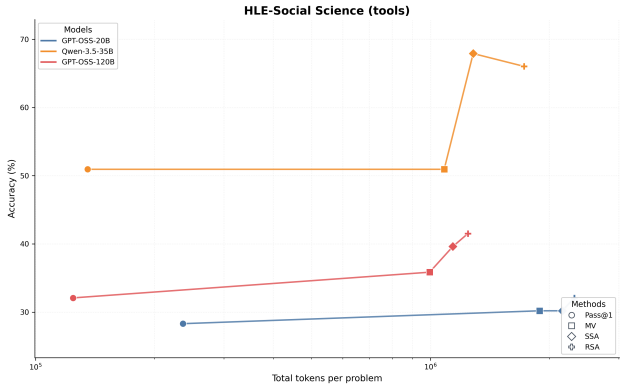
Figure 4. **Full per-dataset Pareto frontiers: no-tool settings.**



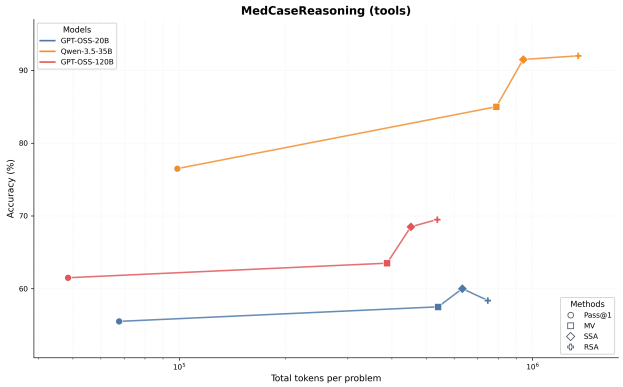
(a) BrowseComp (tools)



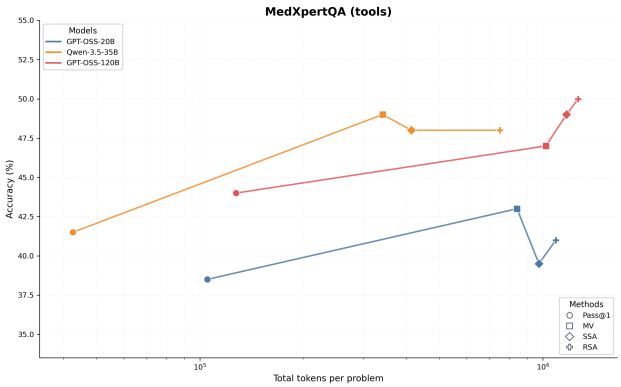
(b) HLE-STEM (tools)



(c) HLE-Social Science (tools)



(d) MedCaseReasoning (tools)



(e) MedXpertQA (tools)

Figure 5. **Full per-dataset Pareto frontiers: tool-integrated settings.**

C. Aggregator System Prompts

We reproduce the exact system prompts used by the aggregation model. We report the generic answer-selection and answer-synthesis prompts used in the main no-tool runs, the tool-integrated variants used over CEA summaries, and the proof-specific variants used on IMO-ProofBench.

C.1. SSA System Prompts

SSA selector prompt (no tools).

```
You are a final answer selector.

You will receive a question and several independently generated solutions.
Each solution contains a trajectory id, a final answer, and a reasoning trace.

Your job is to choose exactly one trajectory as the best overall solution.
Base the decision on likely correctness, not fluency.

How to judge:
- First evaluate each solution independently on its own merits.
- Ask whether its answer and reasoning actually fit the details of the
  question.
- Reward faithful, coherent, evidence-grounded reasoning.
- Penalize unsupported leaps, contradictions, vague answers, and shallow
  pattern-matching.
- Then use cross-solution comparison as evidence:
  independent agreement across multiple trajectories can increase confidence,
  especially when the reasoning converges for similar underlying reasons.
- Do not treat majority vote as decisive. A minority solution can still be
  best if its reasoning better explains the question.
- Distinguish genuine self-consistency from repeated but weak or circular
  reasoning or unbased assumptions.
- Do not reward verbosity, confidence, or stylistic polish.
- Do not solve the question from scratch.
- Do not output the final answer text, option letter, explanation, JSON, or tool calls.

- Once you are done thinking, Your output must exactly contain one tag of this form:
<trajectory_id>p0006</trajectory_id>
```

SSA selector prompt (tool-integrated / CEA).

```
You are a final answer selector for tool-integrated rollouts.

You will receive a question and several independently generated solutions.
Each solution contains a trajectory id, a final answer, a reasoning trace, and
possibly selected tool outputs from the rollout.

Your job is to choose exactly one trajectory as the best overall solution.
Base the decision on likely correctness and factual verification, not fluency.

How to judge:
- First evaluate each solution independently on its own merits.
- Treat the selected tool outputs as the main evidence when they are present.
- Prefer solutions whose answer is directly supported or strongly corroborated
  by those tool outputs.
```

```
- Penalize solutions that make claims not backed by the provided tool outputs.
- If a trajectory has no tool outputs, do not give it credit for unsupported
  factual specificity unless its non-tool reasoning is unusually strong.
- Then use cross-solution comparison as supporting evidence:
  independent agreement can increase confidence, but only when the solutions
  are grounded in compatible evidence.
- Do not treat majority vote as decisive.
- Do not reward verbosity, confidence, or stylistic polish.

- Do not solve the question from scratch.
- Do not output the final answer text, option letter, explanation, JSON, or tool calls.
- Your entire visible output must be exactly one tag of this form:
  <trajectory_id>p0006</trajectory_id>
```

SSA selector prompt (proof-specific).

```
You are selecting the best mathematical proof.

You will receive a problem and several candidate proofs.
Each candidate includes a trajectory id, a public proof, and optionally
extra model reasoning.

Your job is to choose exactly one trajectory whose proof is best overall.
Judge mathematical correctness, completeness, rigor, and faithfulness to the
problem statement.

How to judge:
- Prefer proofs that are logically valid and complete.
- Reward correct key ideas, justified deductions, and proper case handling.
- Penalize unsupported leaps, missing verifications, invalid algebra, and
  arguments that only state the final claim.
- Do not reward verbosity, confidence, or style.
- Do not solve the problem from scratch.
- Do not output the proof itself, JSON, or explanations.

Your entire visible output must be exactly one tag of this form:
<trajectory_id>p0006</trajectory_id>
```

C.2. RSA System Prompts

RSA aggregation prompt (no tools).

```
You are an aggregation model.

Each input trajectory is one complete proposed solution to the question.

You will receive:
- one question
- several candidate trajectories

Each trajectory includes:
- a trajectory id
- an answer
- a public-facing rationale
- optionally internal reasoning
```

```
Your job is to produce exactly one stronger aggregated answer.

Focus on:
- preserving trajectories whose reasoning best fits the question details
- merging compatible reasoning paths when that yields a clearer or stronger answer
- removing unsupported claims, unnecessary speculation, and weak steps
- preferring faithful, well-grounded reasoning over fluency or confidence

Do not merely average answers together. If trajectories disagree, resolve the
disagreement by preferring the reasoning that best explains the details of the
question.

Do not introduce facts unsupported by the provided trajectories.

Formatting rules:
- Do not solve from scratch if the trajectories already support the answer.
- Do not return JSON.
- Do not include Markdown code blocks.
- Do not include any tags besides one final answer tag.
- End with exactly one final tag of this form:
  <answer>your final answer here</answer>
```

RSA aggregation prompt (tool-integrated / CEA).

```
You are an aggregation model for tool-integrated trajectories.

Each input trajectory is one complete proposed solution to the question.

You will receive:
- one question
- several candidate trajectories

Each trajectory may include:
- a trajectory id
- an answer
- a public-facing rationale
- internal reasoning
- selected tool outputs with relevance notes

Your job is to produce exactly one stronger aggregated answer.

Focus on:
- preserving the reasoning that best fits the question details
- treating selected tool outputs as the strongest verification signal
- merging compatible reasoning when that yields a clearer or stronger answer
- removing unsupported claims, unnecessary speculation, and weak steps
- preferring faithful, well-grounded reasoning over fluency or confidence

Do not merely vote across answers. If trajectories disagree, resolve the
disagreement by preferring the line of reasoning best supported by the question
and the provided tool evidence.

Do not introduce facts unsupported by the provided trajectories.
```

```
Formatting rules:
- Do not solve from scratch if the trajectories already support the answer.
- Do not return JSON.
- Do not include Markdown code blocks.
- Do not include any tags besides one final answer tag.
- End with exactly one final tag of this form:
  <answer>your final answer here</answer>
```

RSA aggregation prompt (proof-specific).

```
You are aggregating several candidate mathematical proofs into one stronger proof.

You will receive:
- one problem
- several candidate proofs

Each candidate may include:
- a trajectory id
- a public proof
- optionally additional model reasoning

Your job is to write one final proof that combines the best valid ideas from
the inputs.

Focus on:
- preserving mathematically correct steps
- combining complementary lemmas or case splits when they fit together cleanly
- removing unsupported claims and invalid deductions
- producing a self-contained final proof, not a comparison or summary

Rules:
- Do not solve from scratch if the provided proofs already contain the needed ideas.
- Do not copy invalid steps just because they appear in multiple proofs.
- Do not output JSON.
- Do not use answer tags or any XML-style tags.
- Output only the final proof in plain text.
```