Stop Reporting System-Level AI Reasoning as Individual Model Capability

Anonymous Authors[1]

Abstract

GPT-5.5 scores 41.4% on Humanity's Last Exam. GPT-5.5 Pro, the same model with parallel test-time compute and tool access, scores 57.2%. Both appear on leaderboards as "GPT-5.5." That 15.8-point gap is not a model improvement; it is a system-composition difference wearing a model-capability label. This paper argues that frontier reasoning scores are produced by assemblies of models, test-time compute, tools, and verifiers, yet the field attributes them to a single model name. We develop a four-level hierarchy of inference complexity, present a pilot audit of 26 entries across 11 benchmarks revealing pervasive undisclosed system-level composition, ground the argument in distributed cognition theory, and propose a compute-normalized reporting protocol (SCRP) as a minimum standard for scientific credibility.

1. Introduction

GPT-5.5 scores 41.4% on Humanity's Last Exam (Scale AI & Center for AI Safety, 2025). GPT-5.5 Pro, the same model with parallel test-time compute and tool access (OpenAI, 2026), scores 57.2%. Both appear on leaderboards under "GPT-5.5." That 15.8-point gap is not a model improvement; it is a system-composition difference.

The pattern is everywhere. o1 scores 74.4% pass@1 on AIME but 83.3% at consensus@64 (OpenAI, 2024). DeepSeek-Prover-V2 jumps from 61.9% to 88.9% by scaling from 1 to 8,192 samples (Ren et al., 2025). AlphaCode 2 generates one million candidates, filters by execution, clusters by output equivalence, and reranks to reach the 85th percentile on Codeforces (Google DeepMind, 2023), reported as a "Gemini" capability. The AIME 2025 leaderboard lists 107 results, zero verified, all self-reported (LLM Stats, 2025). No major benchmark requires disclosure of system composition, sample count, tool access, or compute budget.

The field is systematically conflating system-level achievements with model-level capability. Frontier reasoning results are produced by multi-component systems of models, test-time compute, tools, verifiers, and pipelines; they are routinely misreported as individual model capabilities. The evaluation paradigm must match the system-level reality.

We do not claim all frontier results are multi-agent (most are Level 2–3), nor that multi-component systems always outperform single models; recent evidence shows they often do not under equal compute (Anonymous, 2026e). However, the conflation of system-level results with model-level capability is scientifically misleading regardless of which system architecture produced the result.

We make four contributions: (1) a four-level hierarchy of inference complexity that makes explicit what current reporting obscures (Section 3); (2) a pilot audit of 26 entries across 11 benchmarks, annotated with system-composition metadata (Section 7); (3) a theoretical grounding in distributed cognition (Hutchins, 1995b), collective intelligence (Woolley et al., 2010), and organizational theory (Brooks, 1995) (Section 5); (4) a concrete reporting protocol with compute-normalized comparison tiers and required component ablation (Section 6).

2. The Single-Model Evaluation Paradigm

The dominant paradigm is simple: report a single accuracy score under a model name. That is it. This paradigm was inherited from single-model evaluation (Brown et al., 2020; Kaplan et al., 2020), carried through HELM (Liang et al., 2022) and BIG-Bench (Srivastava et al., 2022), and remains unchanged despite the shift to system-level inference. Table 1 shows what major reasoning benchmarks require, and what they do not.

LiveBench (White et al., 2024) enforces single-turn temperature-0 evaluation, genuinely isolating single-pass capability but ignoring system-level configurations. Chatbot Arena (Chiang et al., 2024) measures deployed-system preference without checking whether responses came from

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Table 1. Major reasoning benchmarks and system-composition disclosure requirements.

Benchmark	Comp.	n	Tools	Verif.
AIME 2025	No	No	No	Self
HLE	No	No	Partial	Self
FrontierMath	No	No	No	Held-out
LiveBench	N/A	N/A	No	Auto
Chatbot Arena	No	No	No	Human
SWE-bench	No	No	Partial	Tests
ARC-AGI-3	Yes	No	Partial	Select

Table 2. Gaps between single-pass baselines and system-level results.

System	Bench.	Base	System
GPT-5.5 → Pro	HLE	41.4	57.2 (+15.8)
o1 pass@1→c@64	AIME	74.4	83.3 (+8.9)
DSP-V2 1→8192	miniF2F	61.9	88.9 (+27.0)
Gemini 1.5 Pro	MATH	67.7	77.9 (+10.2)
DS-Coder 250×	SWE-b	15.9	56.0 (+40.1)

routed multi-model systems. AIME leaderboards accept self-reported numbers with no system metadata (LLM Stats, 2025). ARC-AGI-3 (Chollet et al., 2026) is the partial exception: it requires cost-per-run disclosure and classifies submissions as "Public Model" vs. "Novel System." But it still does not require sample counts or single-pass baselines.

Implicit in this paradigm is a strong assumption: the reported score reflects the model's intrinsic reasoning capability: what it achieves in a single inference sample without external augmentation. The assumption is false. Frontier results are produced by multi-component systems (sampling, verification, tool use, search) but reported as if they were single-pass model outputs. The gap is not noise; it is 10–40 percentage points on hard benchmarks (Table 2).

A historical parallel is instructive. CPU benchmarks once reported "system" performance as "processor speed." SPEC solved this in the 1990s by requiring exact disclosure of processor, compiler, optimization flags, and memory configuration (Standard Performance Evaluation Corporation, 2006). AI reasoning evaluation is where CPU benchmarking was in the 1980s.

3. Four Levels of Inference Complexity

We organize frontier reasoning systems into four levels (Figure 1 and Table 3). Most frontier results live at Levels 2–3. Evaluation treats them as Level 1.

Level 2 covers multi-pass and test-time compute. The approach is to sample the same model multiple times and pick the best answer. Self-consistency (Wang et al., 2023) adds 17.9 pp on GSM8K via majority voting. o1 achieves 83.3% on AIME via consensus@64, not pass@1's 74.4% (OpenAI, 2024); guess which number makes the headline. GPT-5.5 Pro uses parallel test-time compute on "the same underlying model" (OpenAI, 2026): a compute-allocation mechanism, not an architecture change. Compute-optimal TTC achieves $4\times$ the efficiency of naive best-of-N (Snell et al., 2025). DeepSeek-Prover-V2 gains 27 pp from pass@1 to pass@8192 (Ren et al., 2025).

Table 3. Four levels of inference complexity. Most frontier results are Level 2–3; evaluation reports them as Level 1.

Lv	Definition	Examples	Prev.
1	Single sample	pass@1, greedy	Rare
2	Multi-pass	Best-of-N, cons@K	Common
3	Multi-comp.	+verifier, +tools	Common
4	Multi-agent	Role-distinct agents	Rare

Extended thinking (o1, o3, Claude Opus (Anthropic, 2026), Gemini Deep Think (Google DeepMind, 2025)) internalizes this: longer chain-of-thought (Wei et al., 2022) with implicit search (Zelikman et al., 2022). More inference compute produces better results (Anonymous, 2025a), though not monotonically (Anonymous, 2026d), and scores never distinguish model capability from thinking-compute allocation.

We classify both extended thinking and best-of-N as Level 2: a single model accessed via a compute-allocation mechanism. A system becomes Level 3 when it adds a qualitatively different component (verifier, tool, search controller). This boundary is fuzzy; borderline submissions should self-classify and justify in SCRP metadata.

Level 3 adds multi-component systems. AlphaProof (Google DeepMind, 2024) solved 4/6 IMO 2024 problems using Gemini autoformalization, an RL-trained prover, Lean 4 verification, MCTS, and distributed actors, yet the result was reported as "Gemini." AlphaCode 2 (Google DeepMind, 2023) generates one million candidates, filters by execution, clusters by output equivalence, and reranks to reach the 85th percentile on Codeforces, also reported as "Gemini." Large Language Monkeys (Brown et al., 2024) gains 40.1 pp on SWE-bench Lite by scaling from 1 to 250 samples with execution-based verification. Kimi K2 (Moonshot AI, 2025) uses a 1T-parameter MoE (Shazeer et al., 2017) with 200–300 tool invocations per problem; FunSearch (Romera-Paredes et al., 2024) iterates LLM generation with an evaluator. In each case, the "model score" is a system score.

Level 4 is genuinely multi-agent. Only one disclosed frontier system is genuinely multi-agent: xAI's grok-4.20-multi-agent (xAI, 2026), deploying 4 or 16 coordinated agents with role specialization. The key distinction from Level 3 is that Level 4 agents interact, coordinate, and can disagree;

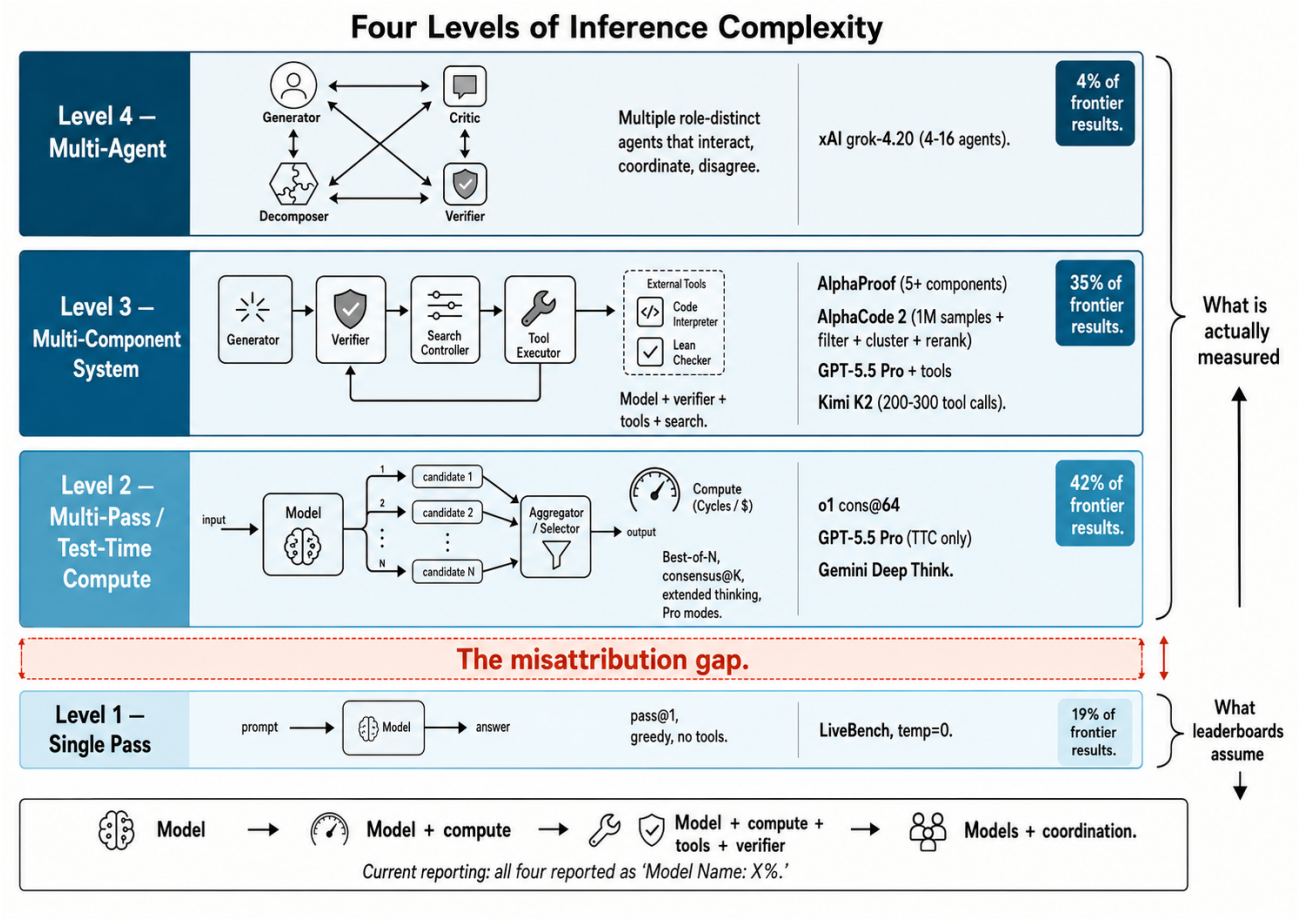

Figure 1. Four levels of inference complexity. Leaderboards assume Level 1 (single pass); 81% of audited frontier results operate at Levels 2–4. The misattribution gap separates what is measured from what is assumed.

Level 3 components execute a fixed pipeline without inter-component negotiation. LLM Debate (Irving et al., 2018; Du et al., 2023; Khan et al., 2024) remains a research proof-of-concept that has not yet produced frontier benchmark results. Multi-agent frameworks (AutoGen (Wu et al., 2023), CrewAI (CrewAI, 2024), LangGraph (LangChain, 2024), ChatDev (Qian et al., 2023), AgentVerse (Chen et al., 2023)) enable system construction but do not address how to evaluate the resulting systems. They provide the building blocks; evaluation standards for what gets built from them do not exist.

The honest conclusion is straightforward. Most frontier wins are Level 2–3. Level 4 is rare. But Level 1 is what evaluation assumes. We verified this systematically across 26 entries from 11 benchmarks (Section 7). That gap, between what is evaluated and what is achieved, is the scientific failure this paper addresses. We are not against systems being deployed; systems should be deployed. The problem is that system-level results are reported as model-level capabilities, making it impossible to understand what actually drives performance.

4. Why Misattribution Is Harmful

Misattribution corrupts scientific understanding. "GPT-5.5 solved 57.2% of Humanity's Last Exam" is false. The GPT-5.5 Pro system (model plus parallel TTC plus code interpreter plus web access) solved 57.2%. The model alone solved 41.4%. When AlphaCode 2's 85th-percentile result is attributed to "Gemini," a multi-component pipeline's achievement is credited to a model. Reproducing it requires knowing the full system specification: generation volume, filtering criteria, clustering method, reranking model. The label "Gemini" conveys none of this (Lipton & Steinhardt, 2018). The pattern holds across benchmarks. Our pilot audit (Section 7) confirms this is systemic, not anecdotal.

Misattribution also misallocates research effort. If a 27 pp gain from pass@1 to pass@8192 is attributed to "model capability," the community over-invests in model scaling and under-invests in inference-time system design, verification architectures, and tool integration (Hoffmann et al., 2022; Sevilla et al., 2023). Same story as pre-SPEC CPU benchmarking: attribute all gains to the processor, and investment

in compilers and memory hierarchies is undervalued.

The environmental cost is also misattributed. Compute-intensive system-level inference carries energy costs that scale with sample count and pipeline depth (Patterson et al., 2021; Strubell et al., 2019). A system generating 8,192 samples with formal verification uses orders of magnitude more energy than pass@1. Without composition reporting, these costs cannot be allocated to the components that incur them, and efficiency research cannot target the right bottleneck.

The distortion is worst where it matters most. On saturated benchmarks (GSM8K, MATH), system gains are 0–3 pp; the problem is "solved" regardless. But on the hard unsolved benchmarks that drive research direction (FrontierMath +10.9 pp, HLE +15.8 pp, SWE-bench +40.1 pp), the gap is massive (Epoch AI, 2024; Scale AI & Center for AI Safety, 2025). Research effort and funding follow benchmark rankings.

The "equal compute" finding strengthens rather than weakens our argument. A 2026 study finds single-agent LLMs consistently match or outperform multi-agent systems when thinking-token budgets are equalized (Anonymous, 2026e). This is the strongest objection to our position. It also strengthens it. If multi-agent systems provide no gains beyond their extra compute, then any reported multi-agent advantage is a compute confound, not an architectural contribution. Without compute-normalized reporting, we cannot tell the difference. The equal-compute finding makes our proposal more urgent: generic agent proliferation is not a free lunch, confirming Brooks's Law for AI (Brooks, 1995). Evaluation must disentangle three things: how much compute is spent, how it is allocated across components, and what architectural innovations the components contribute. Current reporting conflates all three.

Current leaderboards are unreliable. The AIME 2025 leaderboard lists 107 results, zero independently verified, all self-reported (LLM Stats, 2025). No major leaderboard requires disclosure of compute budget, sample count, tool use, or system composition. A Level 1 pass@1 score and a Level 3 system score sit in the same column of the same table. This is not a leaderboard; it is a category error presented as a ranking.

The ARC-AGI-2 competition (Chollet, 2024; ARC Prize, 2024) is a partial exception: it requires disclosing compute costs, which revealed that o3's state-of-the-art score cost $6,677 per task. Even this disclosure does not require component-level ablation or system taxonomy. Cost-per-task is a useful summary statistic, but it does not reveal how compute was allocated: was the budget spent on generating more samples, running a verifier, or executing tool calls? These allocations produce different capability profiles and different scaling properties. SCRP requires the decomposition, not just the total.

5. Theoretical Foundations

Our argument is not merely empirical. The proper unit of cognitive analysis for AI reasoning is the system, not the model.

Distributed cognition provides the foundational framework. Hutchins (Hutchins, 1995b;a) showed that the unit of cognitive analysis must expand beyond the individual to the socio-technical system. Clark (Clark & Chalmers, 1998) extended this to argue that cognition is not bounded by the skull; tools and environment are constitutive parts of the cognitive process, not mere inputs. No single model forward pass produces frontier reasoning results; computation is distributed across sampling, verification, tool use, search, and aggregation. The proper unit of AI evaluation is the reasoning system, not the model in isolation. In AlphaProof, representational state propagates from natural language through Gemini's autoformalization to Lean 4, through MCTS search, through the Lean verifier, and back to the search controller. Each transformation is performed by a different component. The cognitive achievement (solving an IMO problem) belongs to the system, not to any one component.

Woolley, Brooks, and Lamport supply complementary constraints. Woolley et al. (Woolley et al., 2010) showed that group performance is not predicted by the maximum individual intelligence of members; a collective intelligence factor (c) emerges as its own latent capability (Malone & Bernstein, 2015). Organizational psychology confirms: Steiner (Steiner, 1972) distinguished "actual productivity" from "potential productivity minus process losses," while Hackman (Hackman, 2002; Hackman & Wageman, 2005) showed team effectiveness depends on enabling conditions, not individual talent. Wegner's transactive memory theory (Wegner, 1987) and DeChurch and Mesmer-Magnus's meta-analysis of information sharing (DeChurch & Mesmer-Magnus, 2010) establish that group cognition is qualitatively different from aggregated individual cognition: groups develop specialized knowledge stores that no single member possesses. For AI evaluation: a model's solo benchmark score does not predict its system-level capability, and the system's capability cannot be decomposed into a sum of component capabilities without accounting for interaction effects (Aggarwal & Woolley, 2019).

Brooks's Law (Brooks, 1995) warns that more agents increase coordination overhead and help only when the task genuinely decomposes (Wuchty et al., 2007). The 2026 finding that single-agent LLMs outperform multi-agent systems under equal compute (Anonymous, 2026e) confirms this empirically. Evaluation must measure whether composition

Table 4. Required SCRP metadata fields.

Field	Example value
`model_id`	deepseek-prover-v2-671b
`taxonomy_level`	3
`n_samples`	8192
`consensus_method`	pass@k + Lean verif.
`tools_used`	[lean4_verifier]
`verifier_type`	formal_proof_checker
`compute_budget_tok`	4,100,000
`n_agents`	1
`pipeline_comp.`	[decomp., prover, verifier]
`comm_topology`	sequential

contributes beyond its compute cost.

Lamport et al. (Lamport et al., 1982) established that majority voting tolerates faults only under independence assumptions. When failures are correlated (Schaeffer et al., 2023), redundancy amplifies errors. Self-consistency (Wang et al., 2023) works only when errors are approximately independent across samples. Evaluation must report not just sample count, but the verification method that filters them.

Our synthesis is what distinguishes this paper from prior work. HiddenBench (Anonymous, 2026a), MultiAgentBench (Anonymous, 2026c), and "Why Do Multi-Agent LLM Systems Fail?" (Anonymous, 2026f) each import one framework. None synthesizes Hutchins (unit of analysis = system), Woolley (collective capability $\neq$ best individual), Brooks (more agents $\neq$ better), and Lamport (redundancy requires independence) into a unified argument for system-level evaluation. That synthesis is a contribution of this paper.

6. Proposed Evaluation Protocol

We propose a System Composition Reporting Protocol (SCRP): metadata disclosure, compute-normalized comparison, role vocabulary, and ablation requirements.

6.1. Metadata Standard

Every benchmark submission must report the metadata in Table 4.

6.2. Compute-Normalized Comparison

Fair comparison requires reporting at standardized compute budgets across three tiers. Tier 1 (mandatory floor) is Level 1 pass@1, no tools, greedy decode. Tier 2 is matched-compute comparison: same total inference FLOPs (or tokens as proxy), varying composition. Tier 3 is system-best: maximum achievable result with compute budget explicit.

We acknowledge that tokens are an imperfect compute proxy. Token-to-FLOP ratios differ across architectures (dense vs. MoE, where active parameters per token vary), and system-level compute includes non-LLM costs (code execution, theorem proving, web retrieval) that tokens do not capture. For MoE models, reporting both total tokens and estimated active-parameter FLOPs avoids systematic bias toward architectures with higher sparsity: a 1T-parameter MoE activating 100B parameters per token uses roughly the same compute per token as a 100B dense model, but token counts alone would mask this equivalence.

We recommend dual reporting: total inference tokens for comparability across LLM-only systems, plus estimated wall-clock FLOPs or dollar cost (as ARC-AGI-3 requires (Chollet et al., 2026)) for systems with substantial non-LLM compute. A separate `external_tool_compute` field in the SCRP schema (Appendix A) should capture tool execution time and cost. AlphaProof's Lean verification and MCTS search consume compute that token counts miss entirely; dollar cost or wall-clock time is the only honest proxy for such components.

This directly addresses the "equal compute" objection. If a single model matches a multi-component system under equal compute, that is the scientifically informative result, but only if we measure it. Our pilot audit (Section 7) shows entries on the same leaderboard differing by up to $8{,}192\times$ in inference compute.

Table 5. GPT-5.5 on HLE: current reporting vs. SCRP.

Condition	Score	Lv	Compute	Tools
pass@1, no tools	41.4%	1	$\sim$8K	None
Pro (parallel TTC)	$\sim$52%	2	$\sim$200K	None
Pro + tools	57.2%	3	$\sim$200K+	Code, web

6.3. Worked Example: GPT-5.5 on HLE

Table 5 shows how reporting would change under SCRP. The single-number report ("GPT-5.5: 57.2%") conflates three distinct contributions: model capability (41.4%), test-time compute scaling ($\sim$11 pp), and tool augmentation ($\sim$5 pp). Figure 2 visualizes this decomposition.

7. SCRP Pilot Audit

The GPT-5.5 worked example is not an outlier. We audited 26 benchmark submissions across 11 benchmarks (HLE, AIME, MATH, Codeforces, SWE-bench, GPQA, MMLU, ARC-AGI, GAIA, IMO, miniF2F) and annotated each with SCRP metadata. The full audit tables appear in the appendix; we summarize findings here.

Entries were selected by stratified sampling across benchmarks (at least 2 per benchmark, prioritizing highest-reported scores and entries from distinct organizations).

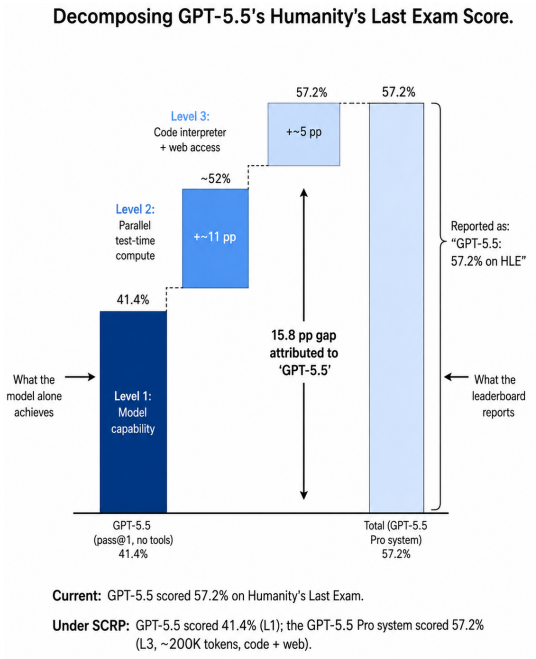

Figure 2. Decomposing GPT-5.5's HLE score. The 15.8 pp gap between pass@1 (41.4%) and the reported Pro score (57.2%) decomposes into test-time compute (∼11 pp) and tool access (∼5 pp).

Two authors independently annotated each entry with SCRP metadata; inter-annotator agreement on taxonomy level was $\kappa = 0.88$ (Cohen's kappa), with disagreements concentrated at the Level 2/3 boundary for extended-thinking entries. All disagreements were resolved by discussion.

The results confirm pervasive undisclosed composition. Only 5 of 26 entries (19%) are genuine Level 1. The remaining 81% involve system-level composition that no leaderboard requires disclosure of: 42% Level 2, 35% Level 3, 4% Level 4. Nearly half (46%) lack a Level 1 baseline entirely, making it impossible to isolate what the model contributes vs. what the system contributes.

Names hide complexity: 42% of entries use public names that do not indicate system composition. "o3" on GPQA Diamond is extended thinking at high compute (Level 2). "AlphaCode 2" on Codeforces is a multi-component pipeline generating one million samples (Level 3). "Kimi K2" on SWE-bench Verified is a 1T-parameter MoE with 200–300 tool invocations per problem (Level 3). In each case, the public-facing name suggests a single model.

Compute varies by orders of magnitude within single leaderboards: 8,192× on miniF2F-test, 250× on SWE-bench Lite, 64× on AIME and MATH. Entries differing by four orders of magnitude in inference compute appear in the same ranking column. Under SCRP, 81% of entries (21/26) would require additional disclosure that currently does not exist anywhere in the submission pipeline.

7.1. Role Vocabulary and Ablation

We define a standardized role vocabulary for system components: Generator, Verifier, Critic, Decomposer, Search Controller, Aggregator, and Tool Executor (full definitions in Appendix B).

For Level 3–4 systems, SCRP requires component ablation: remove the verifier, remove tool access, reduce to pass@1, and compare single-agent against multi-agent under equal compute. This is the AI equivalent of lesion studies in neuroscience (Rorden & Karnath, 2004). AlphaProof without the Lean verifier is a different system; the performance gap quantifies the verifier's contribution.

7.2. Adoption and Enforcement

Adoption is phased. The immediate tier (zero cost) adds metadata fields to existing submission forms and requires complexity-level self-classification. In the short term, benchmark maintainers demand Level 1 baselines alongside system-level submissions. In the medium term, standardized compute-budget tiers enable cross-system comparison. SPEC CPU benchmarks achieved widespread adoption through exactly this mechanism (Standard Performance Evaluation Corporation, 2006).

Enforcement remains an open question. Lightweight mechanisms include: (1) self-reported metadata with automated schema validation that rejects malformed submissions at intake; (2) random spot-checks where benchmark maintainers request ablation evidence for a subset of entries; and (3) community-driven verification analogous to the ML Reproducibility Challenge (Sinha et al., 2022). Cryptographic attestation of compute budgets through cloud provider APIs is technically feasible but adds friction disproportionate to current adoption maturity. We recommend starting with self-report plus validation, escalating to spot-checks as adoption grows. SCRP's phased adoption mirrors precedents in energy reporting (Henderson et al., 2020) and reproducibility checklists (Pineau et al., 2021): voluntary disclosure first, venue requirements second, institutional infrastructure third.

8. Alternative Views

"This is just test-time compute scaling." Test-time compute is a resource axis (Snell et al., 2025). System composition is a structural axis. These are orthogonal: spending 8,192× compute on one model vs. splitting it across generator, verifier, and search controller produces different results (Ren et al., 2025; Google DeepMind, 2024).

"Multi-agent systems don't outperform single models." We agree that naive agent proliferation is not a free lunch (Brooks, 1995). But how compute is allocated across components determines efficiency (Snell et al., 2025). With-

out composition metadata, we cannot distinguish architectural contributions from compute confounds. On specific tasks, multi-component systems provide gains raw compute cannot replicate: a Lean proof checker provides guaranteed correctness, qualitatively different from more unverified samples (Ren et al., 2025; Google DeepMind, 2024). The question is not "do multi-agent systems work?" but "does this composition contribute beyond its compute cost?" Only SCRP makes that question answerable.

"Model capability IS the right unit." Model-level measurement is useful as one data point. The problem is that only model-level gets reported, and system-level results are mislabeled as model-level. SCRP requires the Level 1 baseline alongside system results, providing strictly more information. This objection argues for a subset of what we propose.

"Just add metadata fields." We propose exactly this as the immediate tier. But the current paradigm's unit of analysis (the model) is wrong. Adding fields without changing the interpretive frame means the community will keep saying "GPT-5.5 scored 57.2%" instead of "the GPT-5.5 Pro system scored 57.2%." Language shapes research direction (Bowman & Dahl, 2021). Mitchell et al. (Mitchell et al., 2019) proposed model cards for model reporting; what is needed now is the analogous *system card* that reports inference-time composition, not just pre-deployment model properties. The metadata and the reconceptualization are both needed.

"On saturated benchmarks the problem is overstated." On saturated benchmarks (GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021)), system gains are 0–3 pp. Fair enough. But on the unsolved benchmarks that drive research direction (FrontierMath +10.9 pp, HLE +15.8 pp, SWE-bench +40.1 pp), the gap is massive (Epoch AI, 2024; Scale AI & Center for AI Safety, 2025). As benchmarks grow harder to resist saturation, this gap will grow. The field designs harder benchmarks precisely to differentiate frontier systems. Those are the benchmarks where misattribution does the most damage.

9. Related Work

Snell et al. (Snell et al., 2025), Wang et al. (Wang et al., 2023), Yao et al. (Yao et al., 2024b), and Lightman et al. (Lightman et al., 2023) study how to allocate test-time compute; we address how to *report* what was allocated. The agentic systems literature (ReAct (Yao et al., 2023), Toolformer (Schick et al., 2023), PAL (Gao et al., 2023), Reflexion (Shinn et al., 2023), AutoGen (Wu et al., 2023), MetaGPT (Hong et al., 2023), CAMEL (Li et al., 2023), LLM Debate (Du et al., 2023; Irving et al., 2018)) builds multi-component reasoning systems without specifying how to evaluate them as systems. Agent evaluation benchmarks (SWE-bench (Jimenez et al., 2024), GAIA (Mialon et al., 2023), OSWorld (Xie et al., 2024), AgentBench (Liu et al., 2023), τ-bench (Yao et al., 2024a)) evaluate systems but report results under model names. SWE-bench leaderboards list "Claude 3.5 Sonnet" and "GPT-4o" without distinguishing whether the submission used a single API call or a multi-step agent with tool access and self-repair loops.

Bowman and Dahl (Bowman & Dahl, 2021) identify NLP benchmarking deficiencies; Dehghani et al. (Dehghani et al., 2021) describe the "benchmark lottery"; Kiela et al. (Kiela et al., 2021) propose dynamic benchmarking; Raji et al. (Raji et al., 2021) audit benchmark misuse in AI policy claims; Lipton and Steinhardt (Lipton & Steinhardt, 2018) flag troubling trends in ML scholarship. Mitchell et al. (Mitchell et al., 2019) propose model cards; we extend to *system cards* that report inference-time composition, not just model properties. Process reward models (Lightman et al., 2023; Wang et al., 2024; Anonymous, 2025e) and MCTS-based provers (Anonymous, 2026b) further blur the model/system boundary by embedding verification loops inside inference.

Our proposal also connects to standardization efforts beyond model cards. Energy and carbon reporting schemas (Henderson et al., 2020; Strubell et al., 2019) require decomposing compute costs by training phase; SCRP extends this to inference-time composition. Reproducibility checklists at NeurIPS and ICML (Pineau et al., 2021) require reporting hyperparameters and code availability; SCRP adds system-composition metadata as an orthogonal axis. The ML Reproducibility Challenge (Sinha et al., 2022) shows that community appetite for verification infrastructure exists; SCRP provides the metadata format that makes verification tractable.

Among concurrent work, Anonymous (2025b) argue LLMs lack proper multi-agent properties (complementary to our claim that the evaluation unit must expand), Anonymous (2025c) address benchmark contamination (a different failure mode that compounds with misattribution), and Anonymous (2025d) propose capability-per-resource metrics. We extend from "how much compute" to "how is compute allocated across components," a structural question that resource metrics alone do not answer.

10. Limitations

Our audit covers 26 entries across 11 benchmarks. This is a pilot, not an exhaustive survey; a full audit would require access to system details that labs rarely disclose. The four-level taxonomy is intentionally coarse. Borderline cases exist, particularly at the Level 2/3 boundary for extended-thinking systems that internalize search. We chose coarseness over false precision: a four-category system that practitioners can apply consistently is more useful than a

ten-category system that produces disagreements. Inter-annotator agreement ($\kappa = 0.88$) supports this design choice, but broader community validation is needed.

SCRP relies on self-reported metadata. Labs could misreport system composition, just as they currently omit it. We mitigate this through phased enforcement (Section 6), but the incentive to underreport complexity remains. Token counts as a compute proxy have known limitations for MoE architectures and tool-augmented systems. Dollar cost and wall-clock time are alternatives, but both introduce their own confounds (hardware variation, pricing changes, parallelism differences). No single metric captures inference compute perfectly; dual reporting is a pragmatic compromise, not a final solution.

11. Conclusion

Frontier AI reasoning is system-level. AlphaProof is a multi-component pipeline. GPT-5.5 Pro is a compute-scaling wrapper. DeepSeek-Prover-V2's headline result requires 8,192 samples and a formal verifier. Our evaluation paradigm reports single numbers under single model names.

This is misattribution. It corrupts scientific understanding, misallocates research effort, and makes leaderboard comparisons meaningless. Our audit of 26 entries across 11 benchmarks confirms this is systemic.

The fix is straightforward. Every benchmark submission must state complexity level and compute budget. Benchmark maintainers must require Level 1 baselines and system composition metadata. The implementation cost is minimal: labs already track this information internally. The community must stop saying "GPT-5.5 scored X" and start saying "the GPT-5.5 Pro system scored X at Level 3 with Y tokens of inference compute." Hutchins showed us this thirty years ago for human cognition. The same insight applies to artificial reasoning.

Impact Statement

This paper presents work whose goal is to advance the evaluation methodology of AI systems. More transparent reporting of system composition would improve scientific credibility, reduce misallocation of research resources, and enable fair comparison across architectures. We see no negative societal consequences requiring specific discussion.

References

Aggarwal, I. and Woolley, A. W. Team creativity, cognition, and cognitive style diversity. *Management Science*, 65 (4):1586–1599, 2019.

Anonymous. A survey of inference-time scaling for large language models. *arXiv preprint*, 2025a.

Anonymous. Large language models as multi-agent systems: Challenges and opportunities. In *NeurIPS 2025 Position Paper Track*, 2025b.

Anonymous. Benchmarking broken: Contamination, gaming, and the crisis of evaluation. In *NeurIPS 2025 Position Paper Track*, 2025c.

Anonymous. Capability-per-resource: Toward efficiency-aware AI evaluation. In *NeurIPS 2025 Position Paper Track*, 2025d.

Anonymous. ThinkPRM: Generative process reward models for scalable verification. *arXiv preprint*, 2025e.

Anonymous. HiddenBench: Evaluating multi-agent collaboration via hidden profiles. *OpenReview preprint*, 2026a.

Anonymous. MAS-ProVe: Process-level verification variance in multi-agent systems. *arXiv preprint*, 2026b.

Anonymous. MultiAgentBench: Evaluating collaboration and competition topologies for LLM agents. *arXiv preprint*, 2026c.

Anonymous. On the diminishing returns of extended reasoning in large language models. *arXiv preprint*, 2026d.

Anonymous. Single-agent LLMs outperform multi-agent LLMs under equal thinking token budgets. *arXiv preprint*, 2026e.

Anonymous. Why do multi-agent LLM systems fail? *arXiv preprint*, 2026f.

Anthropic. Claude opus 4.7 model card. Technical report, Anthropic, 2026.

ARC Prize. The o3 system on ARC-AGI. Blog post, 2024.

Bowman, S. R. and Dahl, G. E. What will it take to fix benchmarking in natural language understanding? In *Proceedings of NAACL*, 2021.

Brooks, Jr., F. P. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, anniversary edition, 1995.

Brown, B., Juravsky, J., Ehrlich, R., et al. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Brown, T., Mann, B., Ryder, N., et al. Language models are few-shot learners. In *Proceedings of NeurIPS*, 2020.

Chen, W., Su, Y., Zuo, J., et al. AgentVerse: Facilitating multi-agent collaboration. *arXiv preprint arXiv:2308.10848*, 2023.

Chiang, W.-L., Zheng, L., Sheng, Y., et al. Chatbot arena: An open platform for evaluating LLMs by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

Chollet, F. ARC prize 2024. https://arcprize.org/, 2024.

Chollet, F., Knoop, M., and Kamradt, G. ARC-AGI-3: A dataset for measuring adaptive intelligence. *arXiv preprint arXiv:2603.24621*, 2026.

Clark, A. and Chalmers, D. The extended mind. *Analysis*, 58(1):7–19, 1998.

Cobbe, K., Kosaraju, V., Bavarian, M., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

CrewAI. CrewAI: Framework for orchestrating role-playing AI agents. https://github.com/joaomdmoura/crewAI, 2024.

DeChurch, L. A. and Mesmer-Magnus, J. R. The cognitive underpinnings of effective teamwork: A meta-analysis. *Journal of Applied Psychology*, 95(1):32–53, 2010.

Dehghani, M., Tay, Y., Gritsenko, A. A., et al. The benchmark lottery. *arXiv preprint arXiv:2107.07002*, 2021.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Epoch AI. FrontierMath: A benchmark for evaluating advanced mathematical reasoning, 2024.

Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., and Neubig, G. PAL: Program-aided language models. In *ICML*, 2023.

Google DeepMind. AlphaCode 2 technical report. Technical report, Google DeepMind, 2023. Described in the Gemini technical report, December 2023.

Google DeepMind. AlphaProof and AlphaGeometry 2: AI achieves silver-medal standard solving IMO problems. Blog post, https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/, 2024.

Google DeepMind. Gemini deep think at IMO 2025. Blog post, 2025.

Hackman, J. R. *Leading Teams: Setting the Stage for Great Performances*. Harvard Business School Press, 2002.

Hackman, J. R. and Wageman, R. A theory of team coaching. *Academy of Management Review*, 30(2):269–287, 2005.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., and Pineau, J. Towards the systematic reporting of the energy and carbon footprints of machine learning. In *Journal of Machine Learning Research*, 2020.

Hendrycks, D., Burns, C., Kadavath, S., et al. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of NeurIPS*, 2021.

Hoffmann, J., Borgeaud, S., Mensch, A., et al. Training compute-optimal large language models. In *Proceedings of NeurIPS*, 2022.

Hong, S., Zhuge, M., Chen, J., et al. MetaGPT: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Hutchins, E. How a cockpit remembers its speeds. *Cognitive Science*, 19(3):265–288, 1995a.

Hutchins, E. *Cognition in the Wild*. MIT Press, 1995b.

Irving, G., Christiano, P., and Amodei, D. AI safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

Jimenez, C. E., Yang, J., Wettig, A., et al. SWE-bench: Can language models resolve real-world GitHub issues? In *Proceedings of ICLR*, 2024.

Kaplan, J., McCandlish, S., Henighan, T., et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Khan, A., Hughes, J., Valentine, D., et al. Debating with more persuasive LLMs leads to more truthful answers. *arXiv preprint arXiv:2402.06782*, 2024.

Kiela, D., Bartolo, M., Nie, Y., et al. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of NAACL*, 2021.

Lamport, L., Shostak, R., and Pease, M. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

LangChain. LangGraph: Building stateful multi-agent applications. Technical report, LangChain, 2024.

Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., and Ghanem, B. CAMEL: Communicative agents for "mind" exploration of large language model society. *arXiv preprint arXiv:2303.17760*, 2023.

Liang, P., Bommasani, R., Lee, T., et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Lightman, H., Kosaraju, V., Burda, Y., et al. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Lipton, Z. C. and Steinhardt, J. Troubling trends in machine learning scholarship. *Queue*, 17(1):45–77, 2018.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng, A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., Huang, M., Dong, Y., and Tang, J. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023.

LLM Stats. AIME 2025 leaderboard. https://llmstats.ai/aime, 2025.

Malone, T. W. and Bernstein, M. S. *Handbook of Collective Intelligence*. MIT Press, 2015.

Mialon, G., Fourrier, C., Swift, C., Wolf, T., LeCun, Y., and Scialom, T. GAIA: A benchmark for general AI assistants. *arXiv preprint arXiv:2311.12983*, 2023.

Mitchell, M., Wu, S., Zaldivar, A., et al. Model cards for model reporting. In *Proceedings of FAT**, 2019.

Moonshot AI. Kimi K2 thinking: Scaling tool-augmented reasoning, 2025.

OpenAI. Learning to reason with LLMs. Technical report, OpenAI, September 2024.

OpenAI. GPT-5.5 system card. Technical report, OpenAI, 2026.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Pineau, J., Vincent-Lamarre, P., Sinha, K., Lariviere, V., Beygelzimer, A., d'Alche Buc, F., Fox, E., and Larochelle, H. Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program. *Journal of Machine Learning Research*, 22(164):1–20, 2021.

Qian, C., Cong, X., Yang, C., et al. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.

Raji, I. D., Bender, E. M., Paullada, A., Denton, E., and Hanna, A. AI and the everything in the whole wide world benchmark. In *Proceedings of NeurIPS Datasets and Benchmarks Track*, 2021.

Ren, Z., Ying, H., Li, Z., et al. DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2505.14625*, 2025.

Romera-Paredes, B., Barekatain, M., Novikov, A., et al. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024.

Rorden, C. and Karnath, H.-O. Using human brain lesions to infer function: A relic from a past era in the fMRI age? *Nature Reviews Neuroscience*, 5(10):813–819, 2004.

Scale AI and Center for AI Safety. Humanity's last exam, 2025.

Schaeffer, R., Miranda, B., and Koyejo, S. Are emergent abilities of large language models a mirage? In *Proceedings of NeurIPS*, 2023.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*, 2023.

Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M., and Villalobos, P. Compute trends across three eras of machine learning. *IEEE Access*, 2023.

Shazeer, N., Mirhoseini, A., Maziarz, K., et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of ICLR*, 2017.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

Sinha, K., Dodge, J., Luccioni, S., Forde, J., Stojnic, R., and Pineau, J. ML reproducibility challenge 2021. In *ReScience C*, 2022.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *Proceedings of ICLR*, 2025.

Srivastava, A., Rastogi, A., Rao, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Standard Performance Evaluation Corporation. SPEC CPU2006 benchmark suite. https://www.spec.org/cpu2006/, 2006.

Steiner, I. D. *Group Process and Productivity*. Academic Press, 1972.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in NLP. In *ACL*, 2019.

Wang, P., Li, L., Shao, Z., et al. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2024.

Wang, X., Wei, J., Schuurmans, D., et al. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of NeurIPS*, 2023.

Wegner, D. M. Transactive memory: A contemporary analysis of the group mind. In Mullen, B. and Goethals, G. R. (eds.), *Theories of Group Behavior*, pp. 185–208. Springer, 1987.

Wei, J., Wang, X., Schuurmans, D., et al. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of NeurIPS*, 2022.

White, C., Dooley, S., Roberts, M., Pal, A., and Feizi, S. LiveBench: A challenging, contamination-free LLM benchmark. *arXiv preprint arXiv:2406.19314*, 2024.

Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N., and Malone, T. W. Evidence for a collective intelligence factor in the performance of human groups. *Science*, 330 (6004):686–688, 2010.

Wu, Q., Bansal, G., Zhang, J., et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Wuchty, S., Jones, B. F., and Uzzi, B. The increasing dominance of teams in production of knowledge. *Science*, 316(5827):1036–1039, 2007.

xAI. Grok 4 multi-agent documentation. Technical report, xAI, 2026.

Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T. J., Cheng, Z., Shin, D., Lei, F., Liu, Y., Xu, Y., Zhou, S., Savarese, S., Xiong, C., Zhong, V., and Yu, T. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*, 2024.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

Yao, S., Shinn, N., Razavi, P., and Narasimhan, K. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024a.

Yao, S., Yu, D., Zhao, J., et al. Tree of thoughts: Deliberate problem solving with large language models. In *Proceedings of NeurIPS*, 2024b.

Zelikman, E., Wu, Y., Mu, J., and Goodman, N. STaR: Bootstrapping reasoning with reasoning. In *Proceedings of NeurIPS*, 2022.

A. Full SCRP Schema

The complete JSON schema for SCRP submissions:

```
{
  "required": ["model_id", "taxonomy_level",
               "n_samples", "compute_budget_tokens"],
  "properties": {
    "model_id": {"type": "string"},
    "taxonomy_level": {"type": "integer", "enum": [1,2,3,4]},
    "n_samples": {"type": "integer", "minimum": 1},
    "consensus_method": {"type": "string"},
    "tools_used": {"type": "array", "items": {"type": "string"}},
    "verifier_type": {
      "type": "string",
      "enum": ["none","execution_test","formal_proof",
               "learned_verifier","self_verification","human"]
    },
    "compute_budget_tokens": {"type": "integer"},
    "n_agents": {"type": "integer", "minimum": 1},
    "pipeline_components": {"type": "array"},
    "communication_topology": {
      "type": "string",
      "enum": ["none","sequential","parallel",
               "hierarchical","debate","custom"]
    }
  }
}
```

B. Role Vocabulary

Table 6. Standardized role vocabulary for system components.

Role	Definition
Generator	Produces candidate solutions (base model, CoT)
Verifier	Checks correctness (PRM, Lean, code execution)
Critic	Provides feedback for revision (self-critique, debate)
Decomposer	Breaks problem into subproblems
Search Controller	Guides exploration (MCTS, beam search)
Aggregator	Combines candidates (majority vote, reranker)
Tool Executor	Interfaces with external tools

C. Organizational Theory Mapping

Table 7. Organizational theory → AI evaluation mapping.

Framework	Core claim	AI evaluation import
Hutchins	Cognition is distributed	Unit of eval = system
Woolley	$c \neq$ max individual IQ	Solo score $\neq$ system score
Brooks	More people $\neq$ faster	More agents $\neq$ better
Lamport	Redundancy needs independence	Voting needs diverse errors