CRUCIBLE: A Model-Agnostic Multi-Agent Algorithm for Adversarial Consensus, Citation Integrity, and Provenance-Audited Research with Large Language Models

Jherrod Thomas
Independent Researcher
Jherrod.thomas@aol.com | www.jherrodthomas.com

***Abstract*—Single-model large language model (LLM) pipelines remain vulnerable to hallucination, sycophancy, reasoning shortcuts, and the uncritical reuse of low-quality web content. This paper specifies CRUCIBLE, a model-agnostic orchestration algorithm that forces a heterogeneous ensemble of LLM agents to (i) draft independently, (ii) submit their work to adversarial cross-examination, (iii) cite only from a tier-scored allowlist of primary, peer-reviewed, governmental, or standards-body sources with an explicit blocklist of user-generated-content forums, (iv) reach Byzantine-fault-tolerant super-majority consensus on every atomic claim, and (v) survive a battery of stress tests including adversarial paraphrase, counterfactual injection, and authority swaps. Every claim, critique, retrieval, and vote is written to an immutable provenance ledger that the user may inspect on demand while receiving only the synthesized final answer by default. CRUCIBLE is specified abstractly so that any backend—API-based frontier model, local open-weights model, or a hybrid—can instantiate the generator, critic, judge, and arbiter roles. We position CRUCIBLE against multi-agent debate, Mixture-of-Agents, Chain-of-Verification, Self-Refine, Tree of Thoughts, Constitutional AI, and Byzantine agreement literature, and propose an evaluation protocol grounded in FActScore, TruthfulQA, HELM, and the NIST AI Risk Management Framework.**

Index Terms*—Adversarial evaluation, Byzantine fault tolerance, citation integrity, hallucination mitigation, large language models, multi-agent systems, provenance, retrieval-augmented generation, source credibility, stress testing.*

I. Introduction

LARGE LANGUAGE MODELS (LLMs) trained via next-token prediction on unfiltered web corpora [1], [2] exhibit a well-documented tendency to generate fluent, grammatically correct, and yet factually unsupported statements—a failure mode formalized as *hallucination* [3], [4]. When a single model is asked to perform open-ended research, three failure modes compound: (a) the model invents sources that do not exist; (b) the model quotes real sources that do not support the claim; and (c) the model disproportionately privileges user-generated content such as forum posts, Q&A sites, and encyclopedic stubs over primary, peer-reviewed, or governmental material.

A growing literature addresses individual facets of this problem. Chain-of-Verification [5] prompts a single model to fact-check itself; Self-Refine [6] and Reflexion [7] iterate on a single model's output; Self-Consistency [8] and Tree of Thoughts [9] sample diverse reasoning paths; Multi-Agent Debate [10] and Mixture-of-Agents [11] combine heterogeneous models; Retrieval-Augmented Generation [12] grounds answers in external text; and Constitutional AI [13] introduces rule-based self-critique. Yet no existing pipeline simultaneously enforces (i) cross-model adversarial pressure, (ii) a hard source-quality gate that excludes forum and social-media content, (iii) per-claim entailment verification against a retrieved primary source, (iv) Byzantine-fault-tolerant voting over the ensemble, and (v) a user-inspectable provenance ledger.

This paper specifies CRUCIBLE—**C**ross-model **R**eview **U**nder **C**itation **I**ntegrity with **B**yzantine-tolerant **L**LM **E**nsembling—an algorithm that composes the above components into a single, model-agnostic research pipeline. The user issues a query and, by default, receives a synthesized final answer. On demand, the user may expand the provenance ledger to inspect: each agent's independent draft; every critique and rebuttal; every retrieved source with its tier score; every atomic claim's entailment decision; and every vote. The design target is that a reader who distrusts the final answer can audit the work to the same depth that a peer reviewer would audit a submitted manuscript.

The contributions of this paper are: (1) a formal specification of a nine-stage pipeline that is independent of any particular LLM vendor or weights; (2) a source-gating policy that combines an allowlist of .edu, .gov, standards-body, peer-reviewed-journal, and preprint-archive domains with a blocklist of user-generated-content platforms and a

continuous tier score inspired by the CRAAP heuristic [14]; (3) an adversarial cross-examination protocol that obliges critic agents to submit counter-evidence rather than unsupported dissent; (4) a Byzantine-tolerant arbitration rule adapted from Lamport, Shostak, and Pease [15]; (5) a battery of six pressure and stress tests—adversarial paraphrase, counterfactual injection, temporal perturbation, context-window poisoning, authority swap, and source ablation; and (6) an evaluation protocol that draws on FActScore [16], TruthfulQA, HELM, and the NIST AI Risk Management Framework [17].

Section II surveys related work. Section III formalizes the threat model and design goals. Section IV specifies the nine-stage CRUCIBLE algorithm with pseudocode. Section V describes the system architecture. Section VI proposes an evaluation plan. Section VII discusses threats to validity. Section VIII concludes.

II. Related Work

A. Foundations of LLM Reasoning

The transformer architecture [1] and subsequent alignment techniques such as reinforcement learning from human feedback [2] and Constitutional AI [13] produce models that follow instructions and refuse obvious harms but that remain prone to confident error. Chain-of-thought prompting [18] elicits intermediate reasoning steps, but the steps themselves can be hallucinated.

B. Single-Model Self-Correction

Self-Refine [6] has a model critique and rewrite its own output. Reflexion [7] adds verbal self-reinforcement across trajectories. Chain-of-Verification [5] drafts an answer, plans verification questions, answers each question independently with attention-factoring to avoid bias, and produces a final revised answer. These methods reduce but do not eliminate hallucination, in part because a single model shares its failure modes with its own critic [4].

C. Multi-Path and Multi-Agent Reasoning

Self-Consistency [8] samples diverse chains of thought and marginalizes over the answers. Tree of Thoughts [9] generalizes this into explicit search with self-evaluation and backtracking. Multi-Agent Debate [10] runs multiple model instances that critique and revise one another's answers over several rounds, improving factuality on arithmetic, strategic reasoning, and factual recall. Mixture-of-Agents [11] stacks layers of heterogeneous LLMs, with each agent conditioning on the previous layer's outputs, and reaches state-of-the-art on AlpacaEval 2.0 and MT-Bench using only open-weights models. AutoGen [19] and DSPy [20] supply general-purpose programming frameworks for such pipelines.

D. Retrieval and Source Grounding

Retrieval-Augmented Generation [12] conditions the model on retrieved passages, but the model can still misquote, overgeneralize, or fabricate around real passages. FActScore [16] decomposes long-form answers into atomic facts and scores each against a knowledge base. Provenance tagging [21] attaches a source identifier to each generated span. None of these techniques enforce a quality tier on the retrieved sources themselves; CRUCIBLE's source-gating layer composes with them.

E. Evaluation, Red-Teaming, and LLM-as-Judge

CheckList [22] imports behavioral testing from software engineering. Automated red-teaming [23] generates adversarial prompts to elicit failures. Surveys of LLM-as-judge [24] and of agreeableness bias [25] show that ensemble judging with minority-veto rules outperforms majority vote when individual judges are systematically biased. Dynamic Arbitration for Evaluation (DAFE) [26] uses two primary judges and invokes a third arbitrator only on disagreement, reducing cost without sacrificing reliability.

F. Byzantine Agreement

The Byzantine Generals Problem [15] establishes that, in a system of n nodes where up to f may be arbitrarily faulty, consensus requires $n \geq 3f + 1$. Subsequent work develops practical Byzantine fault tolerance [27]. CRUCIBLE treats the LLM agents as potentially Byzantine participants: an agent may produce fabricated citations, reverse its position under sycophantic pressure, or collude through shared pre-training data. The arbitration rule therefore requires a super-majority rather than a simple majority, and flags any claim on which that threshold is not reached.

G. Governance and Audit

The NIST AI Risk Management Framework 1.0 [17] identifies validity, reliability, accountability, and transparency as core properties of trustworthy AI and names continuous red-team exercises a core safety measure. Recent audit-trail literature [28] argues that agentic AI systems must capture every decision point, tool call, and data access with metadata that explains the context. CRUCIBLE's provenance ledger is designed to satisfy these requirements without overwhelming the end user by default.

III. Problem Formulation

A. Threat Model

We assume an adversary who cannot modify the weights of the participating LLMs but can influence the corpus of retrievable web content (e.g., by seeding a forum post, a typo-squatted domain, or a low-quality encyclopedic edit). We further assume that each individual LLM may: (i) hallucinate facts and citations; (ii) exhibit sycophancy, i.e., agree with whichever claim the prompt appears to favor; (iii) share systematic biases with other LLMs trained on

overlapping corpora [4]; and (iv) pass simple coherence checks while committing subtle logical errors. We do not assume malicious model vendors; however, the algorithm degrades gracefully if up to one-third of the agents in the ensemble are compromised.

B. Design Goals

G1. *Model-agnostic.* Every stage is specified as an interface; any LLM backend that exposes a text-completion endpoint can instantiate a role.

G2. *Show-your-work.* No claim survives to the final answer unless it is attached to at least one primary source that has been automatically checked for (a) URL resolution, (b) membership in the allowlist, (c) absence from the blocklist, and (d) textual entailment of the claim.

G3. *Adversarial pressure.* Every draft must survive critique by at least one model instance that was prompted to disagree and to produce counter-evidence.

G4. *Byzantine tolerance.* Final acceptance of an atomic claim requires a $2f + 1$ super-majority in an ensemble of size $n \geq 3f + 1$.

G5. *User-inspectable provenance.* Every intermediate artifact—prompt, sample, retrieval, critique, vote—is persisted to a ledger keyed by a content-addressable hash, and the user may demand full disclosure at any time.

G6. *Forum-free.* Reddit, Quora, Stack Exchange, personal blogs, content farms, and user-generated wiki content are excluded from the citable source set, regardless of how plausible their content may appear.

IV. The Crucible Algorithm

A. Overview

CRUCIBLE is a nine-stage pipeline. Fig. 1 depicts the dataflow. Stages 1–2 decompose the query and configure retrieval. Stages 3–5 run independent drafts in parallel, expose each draft to adversarial critique, and verify every citation. Stage 6 runs bounded debate rounds. Stage 7 applies Byzantine-tolerant arbitration. Stage 8 stress-tests the draft consensus. Stage 9 synthesizes the final answer and commits the provenance ledger.

B. Stage 1: Query Atomization

Given a user query q, an atomizer agent A_0 decomposes q into an ordered list of atomic, verifiable claim templates $C = \{c_1, \ldots, c_k\}$. An atomic claim is defined, following [16], as a proposition whose truth can be evaluated against a single retrievable source passage. The atomizer also emits a *question type* (definitional, empirical, causal, comparative, procedural, predictive) for each claim, which governs downstream retrieval strategy.

C. Stage 2: Source Gating and Tier Scoring

A deterministic gate G enforces the source policy. Let D denote a candidate source URL. G(D) returns one of $\{\text{ALLOW}, \text{BLOCK}, \text{TIER}_0, \text{TIER}_1, \text{TIER}_2, \text{TIER}_3, \text{TIER}_4\}$. Explicit blocklist patterns (reddit.com, quora.com, stackoverflow.com, *.medium.com, personal blog heuristics, content-farm domains, user-editable wikis) return BLOCK. Remaining sources are scored on five axes adapted from the CRAAP heuristic [14] and from the SIFT framework [29]:

• *Authority*: primary standards body, peer-reviewed journal, governmental agency, or university publication = 4; official technical report = 3; reputable press with editorial oversight = 2; other = 1.

• *Currency*: updated or published within the last five years for fast-moving domains; within fifteen for stable domains.

• *Accuracy*: presence of citations, data, and reproducible methods; cross-confirmation by at least one other TIER_3+ source.

• *Purpose*: research/inform (high) versus persuade/sell (low).

• *Relevance*: measured by a dense retriever score against the atomic claim.

A source is admissible only if its minimum axis score ≥ 2 and its aggregate tier $\geq \text{TIER}_2$. Citable claims in the final answer must rest on at least one TIER_3 or TIER_4 source. The gate G is policy; any organization may substitute a stricter or domain-specific policy without changing the rest of the pipeline.

D. Stage 3: Independent Drafting

N generator agents $\{G_1, \ldots, G_n\}$, instantiated from a heterogeneous pool of LLM backends, answer the query independently. Each generator sees q, the atomic claim set C, and an empty shared context. Critically, generators do not see one another's drafts at this stage; this isolation reduces the collaborative bias identified in [11]. Each generator emits a tuple ⟨draft, claim-ledger, citations, self-confidence⟩. Every sentence in the draft must be mapped to at least one claim in the ledger, and every claim in the ledger must be mapped to at least one citation whose URL has been retrieved through the source-gating layer.

E. Stage 4: Adversarial Cross-Examination

For each draft d_i, M critic agents $\{K_1, \ldots, K_m\}$ are instantiated with prompts that explicitly reward dissent and penalize agreement unsupported by counter-evidence. Critics do not share context with generators or with each other. Each critic emits, for each atomic claim c in d_i, one of: SUPPORTED-WITH-EVIDENCE, CONTRADICTED-WITH-EVIDENCE, UNDER-SUPPORTED (citation does not entail claim), MIS-CITED (source does not exist or does not say what is claimed), or STRONGER-SOURCE-EXISTS (critic proposes a higher-tier citation). Critiques that

merely assert disagreement without supplying a $TIER_2$+ counter-source are discarded; this rule follows Constitutional AI's principle of rule-based self-critique [13] adapted to cross-model critique.

F. Stage 5: Citation Audit and Entailment

An automated citation-audit module V executes, for every claim c and every proposed citation s:

V_1. *URL resolution*: HEAD/GET returns 2xx and content-type matches declared type.

V_2. *Gate check*: $G(s) \geq TIER_2$ and s is not on the blocklist.

V_3. *Passage extraction*: the specific passage quoted or paraphrased is located in the source and returned.

V_4. *Textual entailment*: a natural-language-inference model, or a dedicated entailment prompt run on a held-out LLM, classifies the relation between passage and claim as {ENTAILS, NEUTRAL, CONTRADICTS}. Only ENTAILS is accepted.

V_5. *Uniqueness*: for $TIER_3$ findings the claim must be corroborated by at least one independent $TIER_3$+ source, following the two-source rule in investigative journalism and in FActScore's evidence-retrieval stage [16].

G. Stage 6: Bounded Debate Rounds

For at most R rounds (default R = 3), each generator receives the full set of critiques against its draft plus the citation-audit verdicts and produces a revised draft. Critics then re-critique. A round terminates early if (i) no critic issues a new CONTRADICTED or UNDER-SUPPORTED verdict or (ii) the set of contested claims stabilizes, following the convergence criterion of multi-agent debate [10].

H. Stage 7: Byzantine-Tolerant Arbitration

An arbiter agent J, distinct from all generators and critics, compiles the surviving atomic claims and the votes attached to them. For each claim c, let v(c) denote the number of agents (generators and critics together, totaling n) whose final position SUPPORTS c. Following [15], c is accepted into the consensus set if $v(c) \geq 2f + 1$, where $f = \lfloor (n-1)/3 \rfloor$ is the tolerated Byzantine budget. Claims meeting only a simple majority ($> n/2$) but not the $2f + 1$ threshold are labeled CONTESTED and surfaced in the provenance ledger but not asserted in the final answer. Claims below a simple majority are DROPPED.

I. Stage 8: Pressure and Stress-Test Battery

The consensus set is subjected to six perturbations, each repeated with the full pipeline re-executed on the perturbed input:

T_1. *Adversarial paraphrase*: q is restated by an attacker LLM to maximize semantic equivalence but maximize lexical divergence; the consensus set must be stable under paraphrase.

T_2. *Counterfactual injection*: a false but plausible premise is appended to q; the pipeline must reject the premise rather than inherit it, following the spirit of TruthfulQA.

T_3. *Temporal perturbation*: q is restated with a shifted time window; claims that depend on stale data must be flagged.

T_4. *Context-window poisoning*: a seeded misleading passage is inserted into the retrieval pool; the citation-audit module should reject it.

T_5. *Authority swap*: a high-tier source is replaced with a blocklisted mirror of the same text; the source gate must catch the swap on domain alone.

T_6. *Source ablation*: the single highest-tier source for a claim is removed; the claim should either find an independent corroborating source or demote to CONTESTED.

Claims that fail any stress test are demoted. The battery operationalizes the behavioral-testing philosophy of CheckList [22] and the automated-red-teaming protocol of [23].

J. Stage 9: Synthesis and Provenance Ledger

A synthesizer agent S composes the final answer using only claims that passed Stages 7 and 8. The user receives the synthesized answer, a one-line trust summary ("17 claims, 17 entailed, 14 corroborated by $\geq$ 2 $TIER_3$+ sources, 0 contested after stress tests"), and a single expandable handle to the ledger. The ledger is content-addressed: every prompt, every sampled completion, every retrieved document, every entailment verdict, every vote, and every stress-test result is stored with its SHA-256 hash and a timestamp. A user who opens the ledger can traverse from the final answer down to the exact byte offsets of the cited passage. This design satisfies the audit-trail requirements of [28] and the transparency principle of the NIST AI RMF [17].

Algorithm 1: CRUCIBLE (q, backends, policy)

```
Input: user query q; pool B of LLM backends;
       source policy G; budgets N, M, R, n=N+M.
Output: answer a; ledger L.

C <- Atomize(q)                          // Stage 1
retr <- ConfigureRetriever(C, G)         // Stage 2
for i in 1..N do  in parallel:
    d_i <- Generate(B_i, q, C, retr)     // Stage 3
for i in 1..N, j in 1..M do in parallel:
    k_ij <- Critique(B_{N+j}, d_i, C, retr)   // 4
for each (claim, cite) pair do
    audit <- VerifyCitation(claim, cite) // Stage 5
for round r in 1..R do                   // Stage 6
    d_i  <- Revise(d_i, k_{i*}, audit)
    k_ij <- Critique(B_{N+j}, d_i, C)
    if converged() then break
consensus <- {}
for each atomic claim c do               // Stage 7
    v <- count supporters across N+M agents
    if v >= 2*floor((n-1)/3)+1 then
        consensus <- consensus U {c}
```

```
18 for each test T in {T1..T6} do        // Stage 8
  19   consensus <- StressTest(T, consensus)
20 a <- Synthesize(consensus)            // Stage 9
21 L <- Persist(all prompts, samples,
          retrievals, audits, votes)
22 return (a, handle(L))
```

V. System Architecture

Fig. 2 depicts the reference architecture. The *orchestrator* is a stateless coordinator that dispatches role prompts to an *agent pool* through a thin LLM adapter interface exposing one method: complete(prompt, sampling_params) → text. The adapter admits OpenAI-compatible APIs, Anthropic, Google, and local vLLM or Ollama endpoints without modification. The *source-gating service* is a deterministic policy engine that can be replaced per deployment; an organization with stricter requirements (e.g., legal or medical) substitutes its own rule set. The *retriever* combines a dense neural index over the allowlisted domains with a BM25 fallback and a structured metadata store for publication date, venue, and DOI. The *entailment service* runs a dedicated NLI model and, in parallel, a held-out LLM prompted as a natural-language inference judge; both must agree for ENTAILS. The *ledger* is an append-only log keyed by SHA-256 with optional Merkle-tree commitment for tamper evidence. The *user-facing surface* is deliberately minimal: the final answer, a one-line trust summary, and an "Show work" toggle.

Role assignment is randomized per query with a pinned pseudo-random seed recorded in the ledger. Randomization reduces role-specific conditioning that may arise when a particular backend is always assigned to the same role. The ensemble size n is chosen from the set {4, 7, 10, 13} so that $n = 3f + 1$ exactly; default $n = 7$ tolerates $f = 2$ Byzantine agents.

VI. Evaluation Plan

A. Benchmarks

The pipeline is evaluated on four suites: (i) FActScore [16] on biographical generation for precision of atomic facts; (ii) TruthfulQA [31] for resistance to common misconceptions; (iii) HELM [30] for breadth of capability; (iv) a purpose-built Forum-Free benchmark in which each question has been shown in pilot to elicit at least one Reddit, Quora, or Stack Exchange citation from a baseline single-model pipeline.

B. Metrics

- Atomic-fact precision (fraction of atomic claims in final answer that pass independent human verification);
- Source-tier distribution (fraction of citations at $TIER_3$+);
- Forum-citation rate (target: 0 %);
- Stability under stress tests T_1–T_6 (fraction of claims preserved);
- Calibration (Brier score of the one-line trust summary against human gold);
- Wall-clock latency and token-cost overhead relative to a single-model baseline.

C. Ablations

We plan ablations that independently disable (i) the source-gating blocklist, (ii) the entailment audit, (iii) the Byzantine threshold (replacing with simple majority), (iv) the adversarial critic role, and (v) the stress-test battery. Each ablation is expected to degrade one of the metrics, supporting the claim that the stages are complementary rather than redundant.

VII. Threats to Validity

Several threats merit explicit discussion. *Correlated failure modes.* If all backends share overlapping pre-training data, their "independent" drafts may reproduce the same errors; the Byzantine threshold mitigates but does not eliminate this. Heterogeneity across model families and training-data cut-off dates is a partial defense [11].

Allowlist blind spots. A hard allowlist may exclude emerging but legitimate sources; the tier score is therefore continuous rather than binary, and the ledger surfaces any query whose best available source is below $TIER_3$ so that a human can decide.

Entailment model errors. NLI models remain imperfect; CRUCIBLE therefore cross-checks every entailment verdict with a held-out LLM judge and defers to CONTESTED on disagreement.

Cost. The pipeline is expensive: n generator+critic rounds, R debate rounds, and six stress tests multiply token cost by a factor of roughly $4n \cdot R + 6$. This is acceptable for research-grade answers and prohibitive for high-throughput consumer chat; a deployment may disable the stress battery in low-stakes modes.

Adversarial retrieval. A motivated attacker may seed a blocklisted domain that mirrors a legitimate source; the authority-swap stress test (T_5) specifically targets this.

User trust theatre. A detailed ledger may itself create unjustified confidence. The trust summary therefore reports raw numbers (sources, tiers, contested claims) rather than a single score.

VIII. Conclusion

CRUCIBLE composes known ideas—multi-agent debate, adversarial critique, retrieval-augmented generation, citation auditing, Byzantine agreement, and behavioral stress testing—into a single, model-agnostic research pipeline whose

inputs are a user query and a source policy, and whose outputs are a synthesized answer and an auditable provenance ledger. The pipeline is specified at a level of abstraction that allows any current or future LLM backend to instantiate the generator, critic, judge, and arbiter roles. Future work includes (i) an open-source reference implementation; (ii) a longitudinal study of whether the forum-free constraint changes the demographic coverage of retrieved evidence; (iii) formal guarantees on the soundness of the entailment audit under adversarial retrieval; and (iv) integration with watermarking and model-attestation schemes so that a ledger entry can certify which specific model version produced each intermediate artifact.

References

[1] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008. [Online]. Available: https://arxiv.org/abs/1706.03762

[2] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, New Orleans, LA, USA, Dec. 2022, pp. 27730–27744. [Online]. Available: https://arxiv.org/abs/2203.02155

[3] L. Huang *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–55, 2025, doi: 10.1145/3703155.

[4] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, Mar. 2023, doi: 10.1145/3571730.

[5] S. Dhuliawala *et al.*, "Chain-of-verification reduces hallucination in large language models," in *Findings of the Assoc. Comput. Linguistics (ACL)*, Bangkok, Thailand, Aug. 2024, pp. 3563–3578. [Online]. Available: https://arxiv.org/abs/2309.11495

[6] A. Madaan *et al.*, "Self-Refine: Iterative refinement with self-feedback," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, New Orleans, LA, USA, Dec. 2023. [Online]. Available: https://arxiv.org/abs/2303.17651

[7] N. Shinn *et al.*, "Reflexion: Language agents with verbal reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, New Orleans, LA, USA, Dec. 2023. [Online]. Available: https://arxiv.org/abs/2303.11366

[8] X. Wang *et al.*, "Self-consistency improves chain of thought reasoning in language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Kigali, Rwanda, May 2023. [Online]. Available: https://arxiv.org/abs/2203.11171

[9] S. Yao *et al.*, "Tree of Thoughts: Deliberate problem solving with large language models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, New Orleans, LA, USA, Dec. 2023. [Online]. Available: https://arxiv.org/abs/2305.10601

[10] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," in *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, Vienna, Austria, Jul. 2024. [Online]. Available: https://arxiv.org/abs/2305.14325

[11] J. Wang *et al.*, "Mixture-of-Agents enhances large language model capabilities," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Singapore, Apr. 2025. [Online]. Available: https://arxiv.org/abs/2406.04692

[12] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, Vancouver, BC, Canada, Dec. 2020, pp. 9459–9474. [Online]. Available: https://arxiv.org/abs/2005.11401

[13] Y. Bai *et al.*, "Constitutional AI: Harmlessness from AI feedback," Anthropic, San Francisco, CA, USA, Tech. Rep., Dec. 2022. [Online]. Available: https://arxiv.org/abs/2212.08073

[14] S. Blakeslee, "The CRAAP test," *LOEX Quart.*, vol. 31, no. 3, pp. 6–7, 2004.

[15] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982, doi: 10.1145/357172.357176.

[16] S. Min *et al.*, "FActScore: Fine-grained atomic evaluation of factual precision in long-form text generation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Singapore, Dec. 2023, pp. 12076–12100. [Online]. Available: https://arxiv.org/abs/2305.14251

[17] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST, Gaithersburg, MD, USA, NIST AI 100-1, Jan. 2023, doi: 10.6028/NIST.AI.100-1.

[18] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, New Orleans, LA, USA, Dec. 2022, pp. 24824–24837. [Online]. Available: https://arxiv.org/abs/2201.11903

[19] Q. Wu *et al.*, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," in *Proc. 1st Conf. Lang. Model. (COLM)*, Philadelphia, PA, USA, Oct. 2024. [Online]. Available: https://arxiv.org/abs/2308.08155

[20] O. Khattab *et al.*, "DSPy: Compiling declarative language model calls into self-improving pipelines," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vienna, Austria, May 2024. [Online]. Available: https://arxiv.org/abs/2310.03714

[21] H. Sansford *et al.*, "Provenance: A light-weight fact-checker for retrieval-augmented LLM generation output," arXiv, Nov. 2024. [Online]. Available: https://arxiv.org/abs/2411.01022

[22] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with CheckList," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Jul. 2020, pp. 4902–4912, doi: 10.18653/v1/2020.acl-main.442.

[23] E. Perez *et al.*, "Red teaming language models with language models," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Abu Dhabi, UAE, Dec. 2022, pp. 3419–3448. [Online]. Available: https://arxiv.org/abs/2202.03286

[24] H. Li *et al.*, "LLMs-as-judges: A comprehensive survey on LLM-based evaluation methods," arXiv, Dec. 2024. [Online]. Available: https://arxiv.org/abs/2412.05579

[25] S. Stureborg *et al.*, "Beyond consensus: Mitigating the agreeableness bias in LLM judge evaluations," arXiv, Oct. 2025. [Online]. Available: https://arxiv.org/abs/2510.11822

[26] S. Rahmani *et al.*, "DAFE: LLM-based evaluation through dynamic arbitration for free-form question-answering," arXiv, Mar. 2025. [Online]. Available: https://arxiv.org/abs/2503.08542

[27] M. Castro and B. Liskov, “Practical Byzantine fault tolerance,” in *Proc. 3rd Symp. Oper. Syst. Des. Implement. (OSDI)*, New Orleans, LA, USA, Feb. 1999, pp. 173–186.

[28] Information Systems Audit and Control Association, “The growing challenge of auditing agentic AI,” ISACA, Schaumburg, IL, USA, Industry News, 2025. [Online]. Available: https://www.isaca.org/resources/news-and-trends/industry-news/2025/the-growing-challenge-of-auditing-agentic-ai

[29] M. Caulfield, *Web Literacy for Student Fact-Checkers*. Pressbooks, 2017. [Online]. Available: https://webliteracy.pressbooks.com/

[30] P. Liang *et al.*, “Holistic evaluation of language models,” *Trans. Mach. Learn. Res.*, 2023. [Online]. Available: https://arxiv.org/abs/2211.09110

[31] S. Lin, J. Hilton, and O. Evans, “TruthfulQA: Measuring how models mimic human falsehoods,” in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Dublin, Ireland, May 2022, pp. 3214–3252, doi: 10.18653/v1/2022.acl-long.229.