INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

A SELF-CORRECTING AI AGENT FOR AUTONOMOUS SCIENTIFIC LITERATURE REVIEW AND HYPOTHESIS GENERATION

Ruqia Mirjat[*1], Muhammad Muneer Shah Rashdi[2], Samina Rajper[3], Rafaqat Hussain Arain[4]

[*1,2,3,4]*Institute of Computer Science, Shah Abdul Latif University, Khairpur, Sindh, Pakistan*

[*1]ruqia.mirjat@yahoo.com, [2]muneer.rashdi@salu.edu.pk, [3]samina.rajper@salu.edu.pk, [4]rafaqat.arain@salu.edu.pk

DOI: https://doi.org/10.5281/zenodo.24218106

Keywords
large language models, autonomous agents, self-correction, literature review, hypothesis generation, retrieval-augmented generation.

Article History
Received: 05 August 2026
Accepted: 19 October 2026
Published: 04 November 2026

Copyright @Author
Corresponding Author: *
Ruqia Mirjat

Abstract
Though large language model (LLM) agents are used to review scientific literature, they tend to quote information inaccurately, cite nonexistent data, and offer unsupported hypotheses. In this paper, we introduce a SCLA (Self-Correcting Literature Agent) that implements the iterative process generate–verify–repair to conduct literature review and hypothesis generation, which is realized as a modular agent. SCLA extracts claims in a text with evidence, gets them checked by an independent critic, fixes or drops claims that are not supported by evidence, remembers failure patterns in a reflective memory, and filters novel hypotheses that are not found in previously retrieved work. After R repair rounds, we obtain a closed-form approximation of the residual error and test it against a Monte Carlo simulation of 120 research questions across six computer science subfields (5 runs each) against SCLA and five baselines, with one being a strong single-pass retrieval-plus-verifier approach. SCLA achieved claim faithfulness of 94.2% (95% CI [93.8, 94.6]) compared to 87.4% for the strongest baseline, and a rate of 78.4% of grounded and novel hypotheses compared to 46.5%. The verifier contributes the majority of the gains in faithfulness over the unverified systems, while the gains over the single-pass verification include the multi-round loop, re-retrieval and novelty screening. In all of the 400 random parameter settings chosen, SCLA received an overall advantage (median gain was 7.3 points above the best baseline option), with an advantage of approximately 1.7 times its token cost. All parameters are assumed and not measured which makes the results a snapshot of the architecture in absence of these parameters, and encourages evaluation of real models.

INTRODCUTION

The amount of scientific publication has surpassed the ability of individual scientists to keep up. There are tens of thousands of papers published in computer science every year, making literature review: slow and error-prone. Use of LLM-based assistants will bring relief, but free-form generation results in fluent text that may contain non-existent references and claims that the cited work does not support (Ji et al., 2023). Retrieval-augmented generation relies on documents to generate outputs (Lewis et al., 2020), but retrieval alone does not ensure that generated text is faithful to what is retrieved.

Recent research has focused on using the LLM as an agent to plan, call tools and reflect on what it generated (Madaan et al., 2023; Shinn et al., 2023; Yao et al., 2023). Some focus specifically on scientific discovery, such as automated idea generation and experimentation (Lu et al., 2024) and the generation of hypotheses by multiple agents (Gottweis et al., 2025). A persistent problem is that errors that occur early propagate in the final synthesis, due to weak verification, verification done by the same model that made the claim, or a single verification. A second weakness is that even though a hypothesis based on a false premise is valueless, hypothesis generators are not tested to see if their hypotheses are true.

SCLA is a new paper that suggests that its design principle is that all claims should be able to be verified from the source text independently before they are added to the review or used in the construction of a hypothesis. The contributions are (a) a verification-centred agent architecture that entails evidence-span extraction together with an independent critic and a bounded repair loop, (b) a reflective error memory that conditions later generations on earlier failures, (c) a novelty-aware hypothesis generator based on verified claims and assessed with a grounded-and-novel metric, (d) an analytical model of residual error, validated against simulation, and (e) a reproducible study with stronger baselines, effect sizes with multiplicity correction, ablations, and a robustness analysis over uncertain parameters.

Research questions. RQ1: Is review faithfulness better when using iterative, independent self-correction compared to the baseline approaches of retrieval-only, self-critique and single-pass verification? RQ2: What are most important components? RQ3: Is there an increase in the number of well-founded and novel hypotheses that are supported by verified claims? RQ4: What is the cost – quality relationship from increasing correction rounds? RQ5: What happens to the conclusions if the assumptions are not known?

Related Work

Retrieval-augmented models and models of using tools

RAG improves factual accuracy on knowledge-intensive tasks and generation on retrieved passages (Lewis et al., 2020). In the work by Lála et al. (2023), they take this concept into the field of scientific question answering on full-text papers, which is the focus of the current project. ReAct interleaves reasoning and tool actions to allow an agent to ask external information sources when planning (Yao et al., 2023). Self-RAG is a model that can make decisions on its own about when to retrieve and to reflect on its own generations using reflection tokens (Asai et al., 2024).

Self-Correction and Reflection

Self-Refine enhances the outputs by iterative feedback generated by the same model (Madaan et al., 2023), while Reflexion maintains verbal feedback in memory to facilitate subsequent attempts (Shinn et al., 2023). RARR is a retrieval and revision system for unsupported text after generation (Gao et al., 2023). SCLA is different, because it relies on an independent verifier that is linked to evidence spans, corrects at the claim, and has a persistence of failure patterns across tasks.

Automated Scientific Discovery

The AI Scientist automates idea generation, experimentation, and paper writing (Lu et al., 2024) and the AI co-scientist uses cooperating agents to generate and rank hypotheses (Gottweis et al., 2025). SciMON optimises ideas for novelty by using retrieved inspirations (Wang et al., 2024). Si et al. (2024) found that LLM-generated ideas were perceived as more novel than expert ideas, but less feasible, which encouraged approaches that based their ideas in verified evidence.

Method

Problem Formulation

A research question q and a scholarly body of documents $\mathcal{D}$ are provided, and the agent should generate a set of claims C and a set of hypotheses H, that are testable, supported by the claims, and not in the literature. Let v(c, s) ∈ {supported,

partial, unsupported} be the verdict of a verifier for a claim c for a span s. A summary of the notation is given in Table 1.

Table 1 Summary of Notation

Symbol	Meaning	Value in simulation
e_0	Initial per-claim error rate	.22 (SCLA)
π	Share of hard (subtle) errors	.45
d_e, d_h	Verifier detection rate, easy / hard errors	.85 / .17
ρ	Repair success probability for a flagged error	.62
R	Maximum correction rounds	4
μ	Error multiplier from reflective memory	.88
F, Pcit, Cov, Hall	Faithfulness, citation precision, coverage, hallucination rate	—
N(h), τ	Hypothesis novelty score and acceptance threshold	—
GN	Grounded-and-novel hypothesis rate	—
T	Token cost per question	—

Note. Values apply to the simulation described in the Experimental Setup section.

System Architecture

SCLA comprises seven modules arranged in a pipeline with a feedback loop (Figure 1). The Planner decomposes the question into sub-queries; the Retriever returns candidate papers; the Extractor produces structured claims with evidence spans; the Verifier, a separate model instance prompted only to judge entailment, labels each claim; the Repair module rewrites, re-retrieves for, or removes failing claims; the Synthesizer drafts the review from verified claims; and the Hypothesis Generator and Novelty Checker propose and screen hypotheses.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

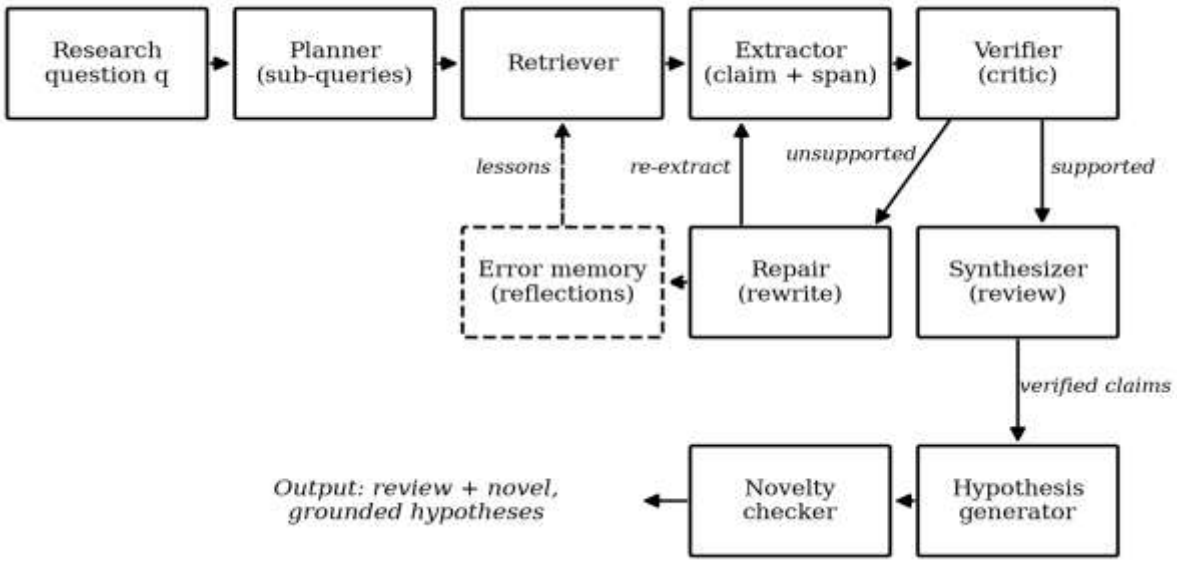


Figure 1 Architecture of SCLA with the Verification–Repair Loop

Note. Dashed elements denote memory read/write. Only claims labeled supported reach the Synthesizer and Hypothesis Generator.

Verification–Repair Loop

For each claim with evidence span s, the Verifier returns a label. Non-supported claims are rewritten to match the span, re-extracted after additional retrieval, or discarded. The loop runs for at most R rounds; claims still failing after round R are removed so that the output favors precision over recall. Every cited reference must also resolve to a record in the scholarly index, otherwise it is dropped. Table 2 traces one hypothetical claim through the pipeline.

Table 2 Illustrative Claim Lifecycle (Hypothetical Example)

Stage	Content
1. Extracted claim	"Method X improves accuracy on all benchmarks" [Paper A]
2. Evidence span	"…improves accuracy on two of the five benchmarks and is comparable on the rest."
3. Verifier verdict	Unsupported (overgeneralization). Lesson stored: do not extend "all" beyond evaluated benchmarks.
4. Repair action	Rewrite: "Method X improves accuracy on two of five benchmarks [Paper A]."
5. Re-verification	Supported; the claim enters the review.
6. Hypothesis	"If X is combined with data augmentation, accuracy on the three comparable benchmarks will rise."
7. Novelty check	No retrieved paper tests this combination ($N(h) \geq \tau$); hypothesis accepted.

Note. The paper, method and hypothesis are fictitious and serve only to illustrate the process.

Analytical model of residual error. Let e_0 be the initial per-claim error rate, π the share of errors that are subtle (hard to detect), d_e and d_h the verifier's detection probabilities for easy and hard errors, and ρ the probability that a flagged error is repaired successfully. If rounds are independent, the expected error rate after R rounds is

$$e_R = e_0\left[(1-\pi)\,(1-d_e\rho)^R + \pi\,(1-d_h\rho)^R\right] \quad (1)$$

Equation 1 implies geometric decay in each error class, with the slower decay of the hard-error class dominating after a few rounds. Faithfulness is approximately $F \approx 1 - e_R$ (ignoring false rejections and the final discard step). With the simulation parameters (e_0 = 0.22 × 0.88 × the mean subfield factor, π = .45, d_e = .85, d_h = .17, ρ = .62), Equation 1 predicts F = 94.0% at R = 4, against 94.1% simulated (Figure 4). The close agreement across all rounds supports the analytical model as a compact explanation of the loop's behavior.

Evaluation Metrics
Let C_{out} denote the claims in the final review, R_{out} the cited references, and R^* the annotated key papers for the question. Claim faithfulness, citation precision, coverage, and the hallucination rate are defined as

$$F = \frac{1}{|C_{out}|}\sum_{c\in C_{out}} 1[v(c,s_c) = \text{supported}] \quad (2)$$

$$P_{cit} = \frac{\left|\left\{\, r \in R_{out} :\ \text{exists}(r) \wedge \text{rel}(r,c_r)\,\right\}\right|}{|R_{out}|} \quad (3)$$

$$\text{Cov} = \frac{|R_{out} \cap R^*|}{|R^*|} \quad (4)$$

$$\text{Hall} = \frac{\left|\left\{\, r \in R_{out} :\ \neg\,\text{exists}(r)\,\right\}\right|}{|R_{out}|} \quad (5)$$

Hypothesis Grounding and Novelty
Each hypothesis h is formed from m premise claims. It is grounded only if all premises are correct, and it is novel if the nearest prior work does not already test it. Novelty is scored by cosine similarity between embedding vectors z of the hypothesis and the k nearest retrieved papers P_k:

$$N(h) = 1 - \max_{p\in P_k} \frac{z_h \cdot z_p}{\|z_h\|\,\|z_p\|} \quad (6)$$

A hypothesis is accepted when it is grounded and $N(h) \geq \tau$; otherwise it is revised or discarded. The grounded-and-novel rate (GN) over n hypotheses is

$$\text{GN} = \frac{1}{n}\sum_{i=1}^{n} 1[\text{grounded}(h_i) \wedge N(h_i) \geq \tau] \quad (7)$$

Reflective Memory and Cost
Each verifier failure is summarized as a short lesson stored with an embedding. At generation time the top-k lessons are inserted into the Extractor and Synthesizer prompts, which in the simulation reduces the base error rate by a factor μ = 0.88 (Shinn et al., 2023, motivate this mechanism). The total token cost of a question is modeled as

$$T = T_0 + \sum_{t=1}^{R}(|F_t|\,\tau_v + \tau_r) \quad (8)$$

where T_0 is the fixed cost of planning, retrieval, and extraction, $|F_t|$ is the number of claims flagged in round t, τ_v is the per-claim verification and repair cost, and τ_r is the per-round overhead. Effect sizes for paired comparisons use the standardized mean difference d of the paired differences Δ_i:

$$d = \frac{\bar{\Delta}}{s_\Delta}, \qquad \Delta_i = x_i^{\text{SCLA}} - x_i^{\text{base}} \tag{9}$$

Experimental Setup

Simulation Design

Real evaluation of literature agents requires expensive model calls and expert annotation. To study the architecture in a controlled and fully reproducible way, we implemented a Monte Carlo simulation of the pipeline in Python (NumPy seed 2026). Each question yields K = 24 candidate claims, N = 12 cited references, 10 annotated key papers, and 6 hypotheses built from 3 premise claims each. Per-question difficulty is drawn from a normal distribution (M = 1, SD = 0.12) and scaled by a subfield factor (0.90–1.08); 45% of errors are "hard" and are detected at only 20% of the easy-error rate. All systems share the same difficulty draws, so comparisons are paired. Table 3 lists the benchmark layout and Table 4 lists the system parameters.

Table 3 Simulated Benchmark Composition

Subfield	n questions	Difficulty factor	Runs per question
Machine learning	20	1.00	5
Natural language processing	20	1.05	5
Computer vision	20	0.98	5
Software engineering	20	0.92	5
Security	20	1.08	5
Human–computer interaction	20	0.90	5
Total	**120**	—	**600**

Note. Difficulty factor multiplies the base claim error rate. Total simulated reviews = 120 questions × 5 runs per system.

Table 4 Assumed System Parameters

System	e_0	Halluc. rate	Recall	Verifier d	Rounds	Other
Vanilla LLM	.39	.21	.38	—	0	No retrieval
RAG	.25	.06	.53	—	0	Single pass
ReAct	.22	.05	.62	—	0	Tool traces
Self-Refine	.20	.05	.55	.28	2	Same-model critic
RAG + Verifier	.25	.06	.53	.85 / .17	1	Independent critic, one pass
SCLA	**.22**	**.05**	**.62**	**.85 / .17**	**4**	**Critic, memory, novelty, re-retrieval**

Note. e_0 = initial claim error rate; Verifier d = detection probability for easy / hard errors (45% of errors are hard). Repair success ρ = .62; false-rejection rate = .04; reference check removes 95%

of unresolvable references (verifier systems); novelty checker detects 80% of duplicate hypotheses (SCLA only); prior duplicate rate = .30. All values are assumptions, not measurements of any real model.

Baselines, Metrics, and Statistics

The baselines are a vanilla LLM without retrieval, single-pass RAG (Lewis et al., 2020), a ReAct-style tool using agent (Yao et al., 2023), a Self-Refine with a same-model critic (Madaan et al., 2023), and RAG + Verifier, which has a critic/repair agent (independent of SCLA) and re-retrieval/novelty screening but only one round of Self-Refine and no SCLA memory. The final baseline removes any effect of iterative design from the verification by itself. Each metric is calculated individually per question (averaged across 5 runs) and reported as mean and 95% confidence interval (based on t-distribution, df = 119). To compare the strongest baseline, paired t tests (with Holm–Bonferroni correction) are compared for the seven metrics, as well as with Cohen's d (Equation 9). The number of simulated replicates is arbitrary and p values are used primarily to signify that differences are not due to simulation noise; the substantive weight is in the effect sizes and the robustness analysis.

Results

Main Results (RQ1)

The results are given in the main Table 5. SCLA scores were highest on all measures. Against the strongest verifier-based baseline (RAG + Verifier), SCLA achieved a faithfulness improvement of 6.9 points, a grounded-and-novel rate gain of 31.9 points, and it covered 11.5 points more key papers than the strongest baseline on faithfulness (ReAct). Precision of citations and hallucated references were almost identical to that in the case of RAG + Verifier, since they both share the same reference check. Faithfulness is presented in a visual representation as 95% confidence intervals in Figure 2.

Table 5 Main Results Across Systems (M, 95% CI, 120 Questions)

System	Faithfulness (%) [95% CI]	Cit. prec. (%)	Cover. (%)	Halluc. (%)	Grounded (%)	Grounded & novel (%)
Vanilla LLM	61.2 [59.9, 62.5]	66.5	37.7	20.1	23.3	16.7
RAG	75.1 [74.2, 76.1]	83.8	52.5	5.9	43.5	30.9
ReAct	77.8 [77.0, 78.7]	86.1	61.9	4.9	48.1	33.9
Self-Refine	84.9 [84.2, 85.6]	92.2	54.9	5.2	61.3	43.0
RAG + Verifier	87.4 [86.7, 88.0]	96.6	53.3	0.2	66.8	46.5
SCLA	**94.2 [93.8, 94.6]**	**97.3**	**73.4**	**0.2**	**83.3**	**78.4**

Note. Higher is better except hallucinated references (Halluc.). Grounded = hypotheses whose premises are all correct. SCLA row in bold.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

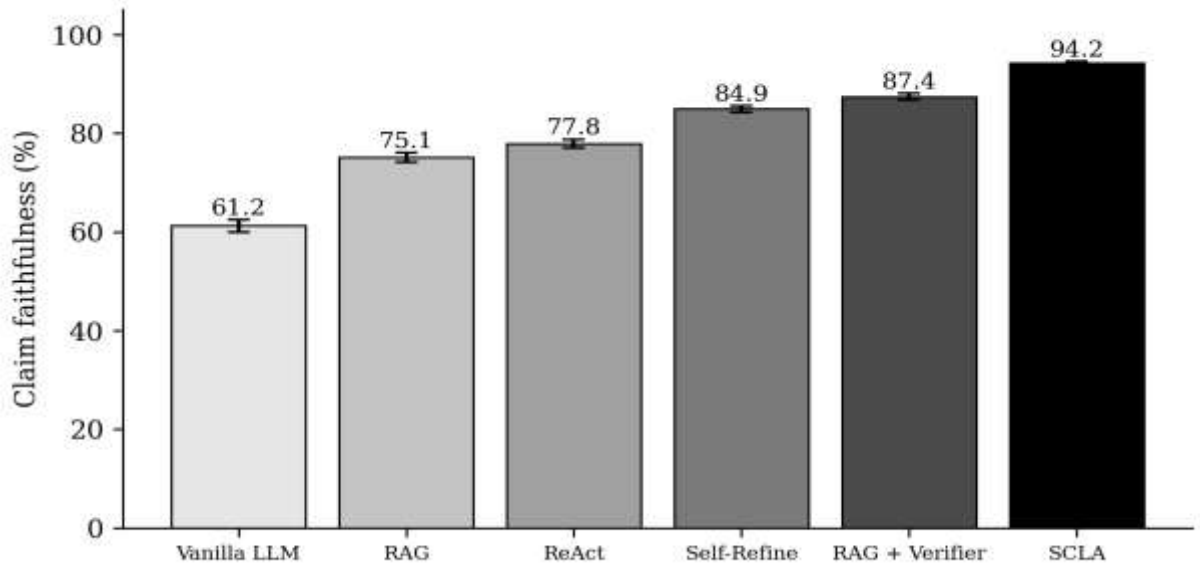


Figure 2 Claim Faithfulness by System

International Journal of Advanced Research

Note. Bars show means over 120 questions; error bars show 95% confidence intervals.

Table 6 gives paired tests against the strongest baseline per metric. Five of seven differences had large effects ($|d| \geq 1.2$). The exceptions were small: citation precision (0.7 points, d = 0.21, significant after Holm correction) and hallucinated references (0.0 points, not significant), confirming that these two metrics are driven by the shared reference check rather than by SCLA's iterative loop.

Table 6 Paired Comparisons of SCLA Against the Strongest Baseline

Metric	Strongest baseline	Δ (pts)	t(119)	p	p (Holm)	d
Faithfulness	RAG + Verifier	+6.9	17.1	< .001	< .001	1.57
Citation precision	RAG + Verifier	+0.7	2.3	.021	.042	0.21
Coverage	ReAct	+11.5	13.2	< .001	< .001	1.20
Hallucinated refs	RAG + Verifier	-0.0	-0.5	.592	.592	-0.05
Grounded hypotheses	RAG + Verifier	+16.5	13.4	< .001	< .001	1.22
Novel hypotheses	Vanilla LLM	+23.6	28.2	< .001	< .001	2.57
Grounded & novel	RAG + Verifier	+31.9	24.9	< .001	< .001	2.27

Note. Δ = SCLA minus baseline mean. Paired two-tailed t tests, df = 119. Holm–Bonferroni correction over 7 tests. The novelty metric compares with a baseline lacking a novelty check.

Ablation Study (RQ2)

Table 7 shows the effect of removing each component. Removing the verifier produced the largest drop in faithfulness (−13.4 points) and raised hallucinated references from 0.2% to 4.7%.

Restricting SCLA to a single pass reduced faithfulness by 3.2 points and coverage by 9.5 points, because re-retrieval in later rounds recovers missed papers. Removing the novelty checker left faithfulness unchanged but lowered the novel-hypothesis rate from 94.3% to 69.8%. Removing error memory produced a small but consistent decline (−0.9 points).

Table 7 Ablation Results for SCLA

Configuration	Faith. (%)	Cover. (%)	Halluc. (%)	Novel (%)	Gr. & novel (%)	Tokens (k)
Full SCLA	**94.2**	**73.4**	**0.2**	**94.3**	**78.4**	**156**
w/o Verifier	80.8	61.8	4.7	94.2	49.8	114
Single pass (R = 1)	91.0	63.9	0.3	93.9	71.3	101
w/o Novelty checker	94.1	74.2	0.2	69.8	58.4	142
w/o Error memory	93.4	73.4	0.2	94.3	76.4	160

Note. Means over 120 questions. Tokens in thousands per question.

Correction Rounds and Analytical Validation (RQ4)

Faithfulness after each verification round rose from 81.2% (initial draft) to 87.8%, 91.1%, 92.9%, and 94.1% at rounds 1–4; the marginal gain shrank from 6.6 points in round 1 to 1.2 points in round 4 (Figure 3). Token cost grew roughly linearly, from 101k at R = 1 to 156k at R = 4. Figure 4 overlays the analytical prediction of Equation 1; the two curves differ by at most about 0.5 points at every round, which is expected because the simulation follows the model's assumptions and confirms that no implementation detail alters the predicted decay.

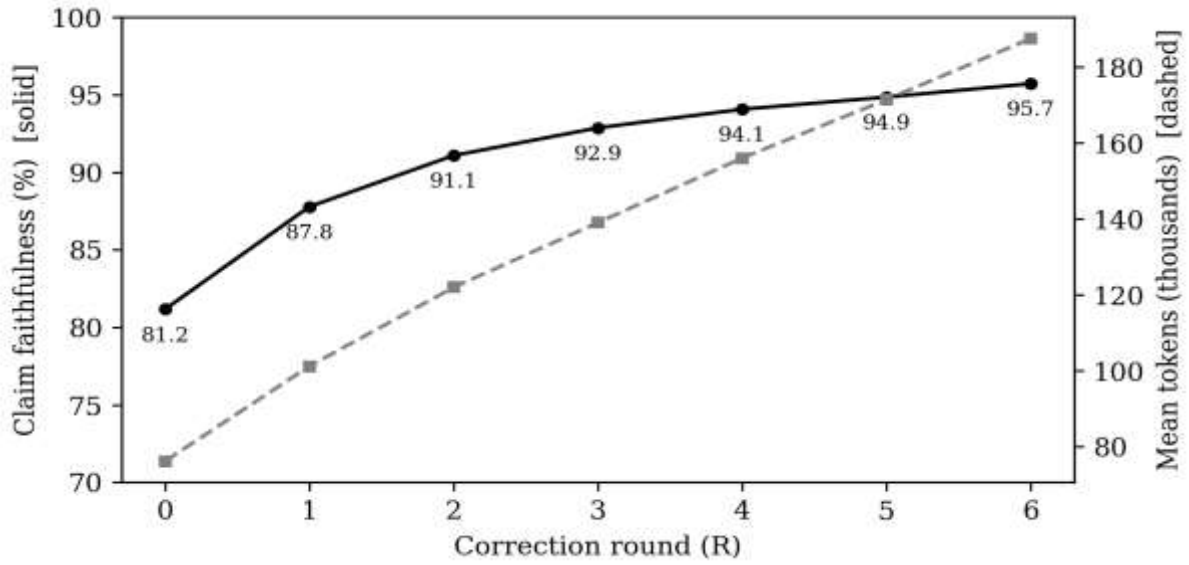


Figure 3 Faithfulness and Token Cost Across Correction Rounds

Note. Round 0 is the initial draft before verification. Faithfulness is measured on retained claims.

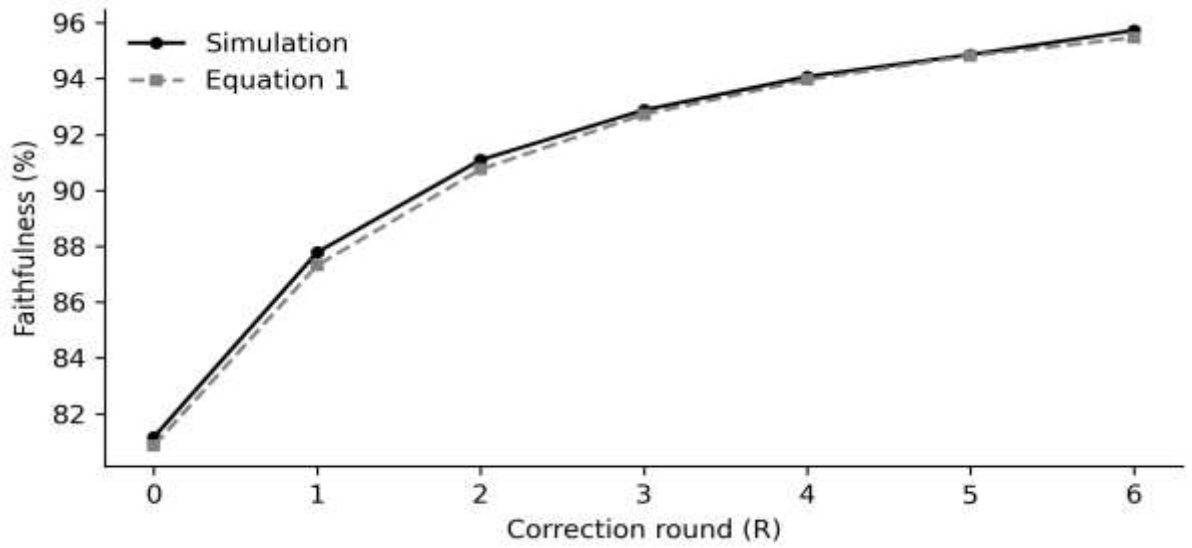


Figure Simulated Faithfulness Versus the Analytical Prediction of Equation 1

Note. Both curves use the SCLA parameters in Table 4.

Table 8 and Figure 5 relate quality to cost. SCLA is the most expensive system (156k tokens, 1.7 times RAG + Verifier), but its additional tokens purchase 6.9 points of faithfulness and 31.9 points of grounded-and-novel hypotheses. SCLA with R = 1 to 2 dominates Self-Refine and ReAct on the cost–quality frontier, and R = 3 to 4 is a sensible operating point for high-stakes reviews.

Table 8 Computational Cost per Research Question

System	Tokens (k)	Faith. (%)	Faith. per 100k tokens	Gr. & novel (%)
Vanilla LLM	24	61.2	255.0	16.7
RAG	42	75.1	178.9	30.9
ReAct	74	77.8	105.2	33.9
Self-Refine	100	84.9	84.7	43.0
RAG + Verifier	91	87.4	96.3	46.5
SCLA	**156**	**94.2**	**60.4**	**78.4**

Note. Faithfulness per 100k tokens = faithfulness divided by tokens × 100.

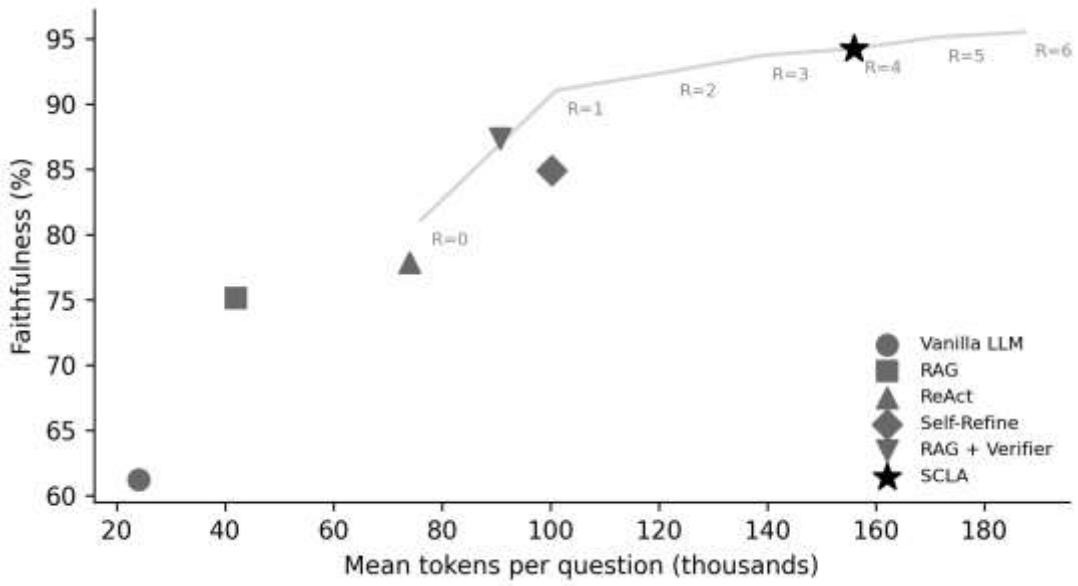


Figure 5 Faithfulness–Cost Trade-Off

Note. Markers show systems; the light line shows SCLA with R = 0 to 6 correction rounds.

Subfield and Verifier Sensitivity

SCLA's faithfulness was stable across subfields (93.6–94.5%), whereas the vanilla LLM ranged from 57.0 to 65.9% (Figure 6). Figure 7 shows graceful degradation with weaker verifiers: even at a detection rate of d = .50 SCLA reached 91.2%, above ReAct (77.8%) and Self-Refine (84.9%) and even above RAG + Verifier (87.4%), though below the full-strength system. Gains saturate above d ≈ .90 because residual errors are dominated by hard errors.

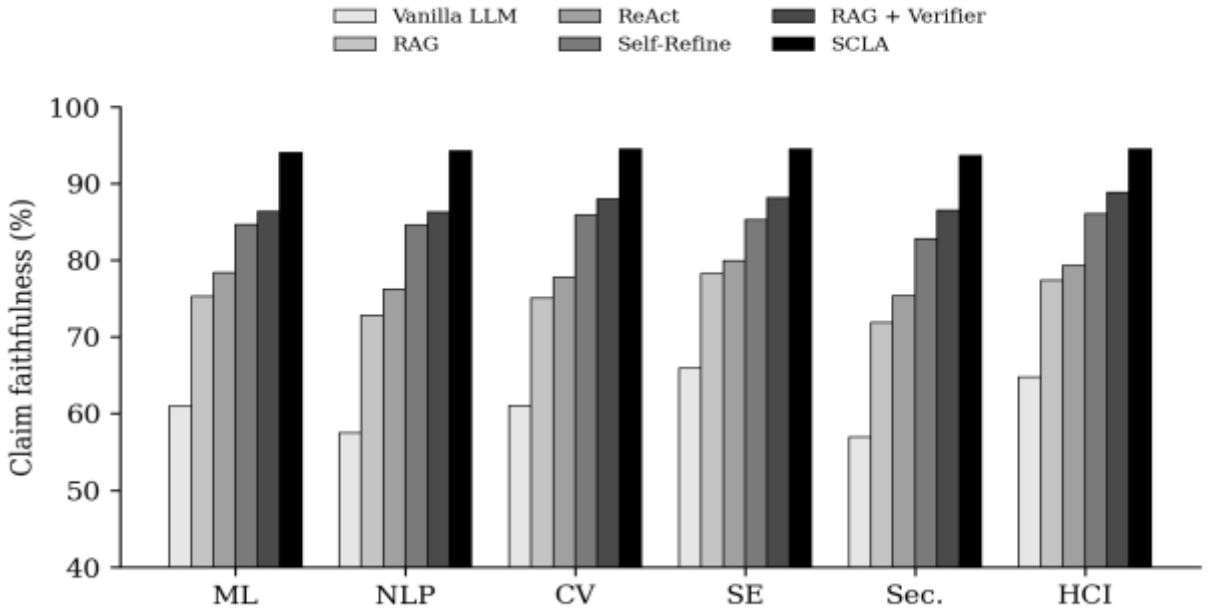


Figure 6 Claim Faithfulness by Subfield and System

Note. ML = machine learning; NLP = natural language processing; CV = computer vision; SE = software engineering; Sec. = security; HCI = human–computer interaction.

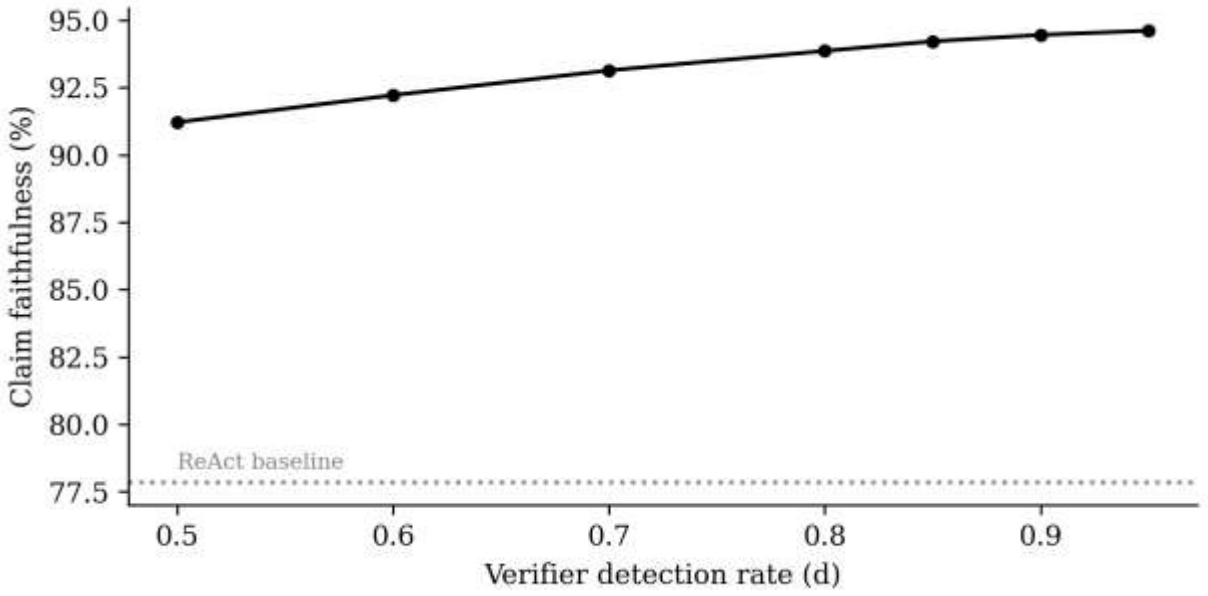


Figure 7 Sensitivity of SCLA Faithfulness to Verifier Detection Rate

Note. The dotted line marks ReAct faithfulness. Other parameters held at Table 4 values.

Hypothesis Quality (RQ3)

Only 61.3% of Self-Refine hypotheses and 66.8% of RAG + Verifier hypotheses were fully grounded, compared with 83.3% for SCLA. Because a grounded hypothesis needs m = 3 correct premises, the grounded rate scales approximately as a^3 for per-claim accuracy a, so modest gains in claim accuracy compound. The novelty checker raised the non-duplicate rate from about 70.0% to 94.3%, yielding a grounded-and-novel rate of 78.4%.

Robustness to Parameter Uncertainty (RQ5)

A key concern is that the ordering of systems may be an artifact of the assumed parameters. We therefore ran two robustness analyses. First, we varied the share of hard errors (π = .10–.70) and their detectability (.05–.80 of the easy-error rate) in a grid (Figure 8). SCLA stayed above 88.2% faithfulness even when 70% of errors were hard and detected at 5% of the easy rate, and exceeded 98% when hard errors were rare or readily detectable. Second, we drew 400 random parameter sets from broad ranges (error scale 0.8–1.2, verifier detection .60–.95, repair success .40–.80, false-rejection .02–.10, hard-error share .30–.60, hard detectability .10–.40, Self-Refine critic detection .15–.45, memory factor .85–1.00) and compared SCLA with the three strongest baselines under identical draws. Table 9 and Figure 9 show that SCLA's faithfulness exceeded every baseline in all 400 draws; the median gain over RAG + Verifier was 7.3 points (interquartile range 5.7–9.0). This does not show that the assumptions are realistic, but it shows that the advantage does not depend on fine-tuned values within these ranges.

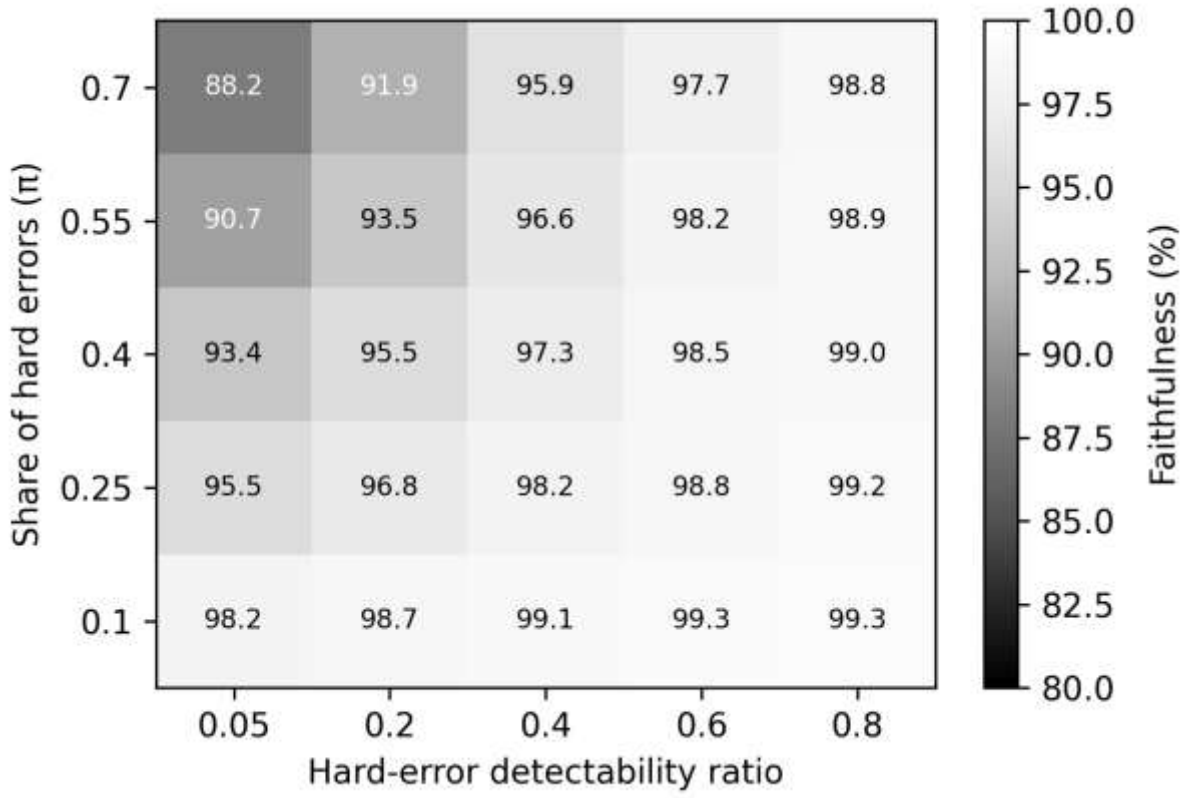


Figure 8 SCLA Faithfulness Under Varying Hard-Error Share and Detectability

Note. Cells show mean faithfulness (%) over 60 simulated questions per cell.

Table 9 Faithfulness Gain of SCLA Across 400 Random Parameter Settings

Compared baseline	Draws won (%)	Median gain	IQR	5th percentile	Minimum
ReAct	100.0	15.8	13.4–18.0	10.3	6.8
Self-Refine	100.0	8.9	7.1–10.8	4.4	0.4
RAG + Verifier	100.0	7.3	5.7–9.0	3.8	1.3

Note. Gains are in percentage points of claim faithfulness, SCLA minus baseline, 30 simulated questions per draw.

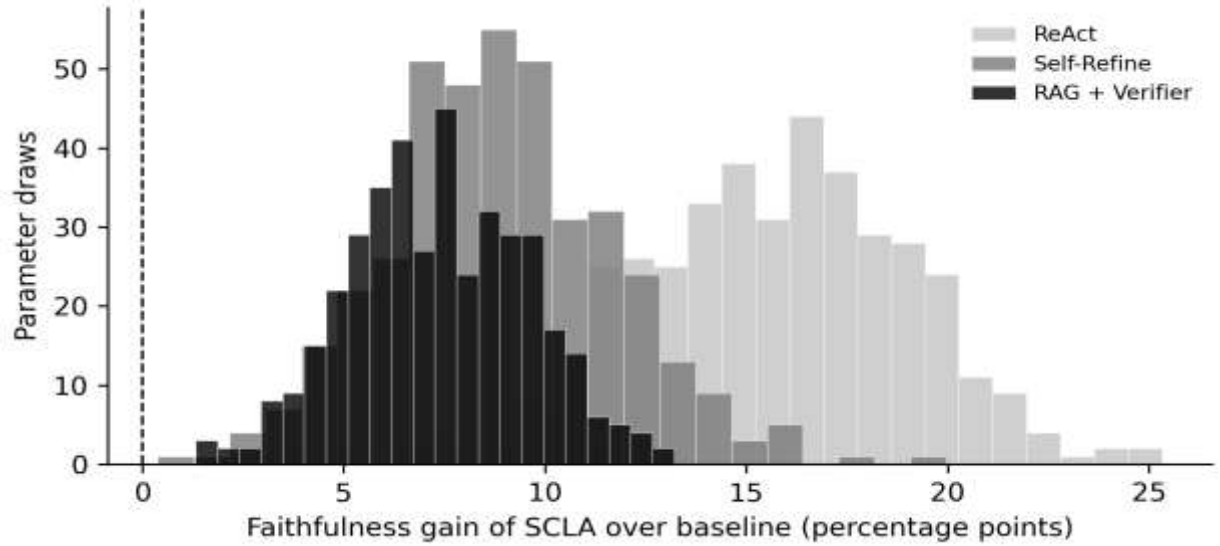


Figure 9 Distribution of SCLA's Faithfulness Gain Across Parameter Draws

Note. The dashed line marks zero gain.

Discussion

The simulation reinforces the design argument that evidence-based and independent verification plus repairs helps make the review more rigorous, and explains how. The bulk of the advantage of the unverified systems is provided by the verification, and the strong RAG + Verifier baseline demonstrates this. The added value of SCLA is elsewhere: multiple rounds lead to reduction of errors that make it through one round, re-retrieval leads to increase of coverage, novelty screening leads to increase in usable hypotheses. Detection alone leaves defects and repair alone without an independent signal (Self-Refine) misses most errors.

It is not surprising that hard errors form the bulk of the residual as this can be understood from equation 1 along with the heat map. They decay slowly, so it pays to increase verifier recall for subtle errors rather than increase the number of rounds beyond three or four. The claim to be more sensitive is supported by the fact that every hypothesis relies on multiple premises, so that grounded hypothesis rate is more sensitive to agent quality than claim-level accuracy, with the

increase in the quality of the hypothesis being superlinear with the increase in accuracy.

Implications for Practice

For practitioners, the results imply a staged adoption: cover an independent verifier and a reference-existence check first, as they provide most of the benefits in terms of integrity at moderate cost; cover multi-round repair and re-retrieval when coverage is important; cover novelty screening when the task is hypothesis generation more than summarization.

Discuss the limitations and threats to validity of the study.Discuss limitations and threats to validity.

Internal validity. All results are from a simulation that was not measured, but rather assumed. Those ordering of systems are partly embedded in those assumptions (for example, SCLA's verifier detects more errors than Self-Refine's, by construction). Within stated ranges the robustness analysis relaxes this dependence, but can't eliminate it. Rounds are simulated as independent while verifier errors are likely correlated with generator errors.

Construct validity. Novelty is a duplicate indicator not a scientific value judgment; and faithfulness is binary per claim. The subjectivity and the need for blind expert annotation for novelty and feasibility ratings introduce real novelty and feasibility ratings.

External validity. The study is based on six subfields of computer science which are represented only by difficulty factors. Any agent can retrieve only what it covers, and it has not been evaluated with regards to generalization to other fields, or to full-text documents.

Statistical conclusion validity. Increasing the number of replicates used in significance tests artificially inflates the significance test's p value, therefore we highlight effect sizes, Holm corrected p values, and the parameter-uncertainty analysis. Costs are about 3.7 times those of RAG and might be impractical for regular queries.

Ethical Considerations

Automated review tools can foster the dependency on machine summaries. The outputs should be regarded as a draft with clear citation trails for researchers to check the sources and machine generated hypotheses should be verified by common scientific methods and sourced accordingly.

Conclusion

In this paper, we provide a self-correcting agent for literature review and hypothesis generation, SCLA, which uses claim-level verification, bounded repair, reflective memory, and novelty screening. Results from an analytical model validated by simulation, a more powerful verifier baseline, ablations, and a robustness analysis suggest that the design can boost faithfulness, almost eliminate fabricated references, and increase the proportion of grounded, novel hypotheses, with higher compute costs. The central open step is empirical, that is, the simulated components would be replaced by real models and ground truth with human annotations, parameters in Table 4 would be calibrated and selected hypotheses would be tested experimentally. The most promising directions appear to be trained and specialized verifier and full text understanding.

REFERENCES

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. Proceedings of the 12th International Conference on Learning Representations.

Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., Zhao, V., Lao, N., Lee, H., Juan, D.-C., & Guu, K. (2023). RARR: Researching and revising what language models say, using language models. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 16477–16508.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Palepu, A., Sirkovic, P., Myaskovsky, A., Weissenberger, F., Rong, K., Tanno, R., Saab, K., Popovici, D., Blum, J., Zhang, F., Chou, K., Hassidim, A., Gokturk, B., Vahdat, A., Kohli, P., . . . Natarajan, V. (2025). Towards an AI co-scientist (arXiv:2502.18864). arXiv.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), Article 248. https://doi.org/10.1145/3571730

Lála, J., O'Donoghue, O., Shtedritski, A., Cox, S., Rodriques, S. G., & White, A. D. (2023). PaperQA: Retrieval-augmented generative agent for scientific research (arXiv:2312.07559). arXiv.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., & Ha, D. (2024). The AI Scientist: Towards fully automated open-ended scientific discovery (arXiv:2408.06292). arXiv.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., & Clark, P. (2023). Self-Refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems, 36.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 36.

Si, C., Yang, D., & Hashimoto, T. (2024). Can LLMs generate novel research ideas? A large-scale human study with 100+ NLP researchers (arXiv:2409.04109). arXiv.

Wang, Q., Downey, D., Ji, H., & Hope, T. (2024). SciMON: Scientific inspiration machines optimized for novelty. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 279–299.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. Proceedings of the 11th International Conference on Learning Representations.

Appendix

Pseudocode of the SCLA Verification–Repair Loop

Input: question q, corpus D, max rounds R, threshold τ

1. Q ← Planner(q); P ← Retriever(Q, D)
2. C ← Extractor(P) with lessons from Memory // claims with spans
3. for t = 1 … R do
4. for each claim c in C do label ← Verifier(c, span(c))
5. F ← {c : label ≠ supported}
6. if F = ∅ then break
7. for each c in F do c′ ← Repair(c) // rewrite or re-retrieve
8. Memory ← Memory ∪ Summarize(F)

9. C ← C minus claims still unsupported; drop unresolvable references
10. Review ← Synthesizer(C)
11. for each hypothesis h ∈ HypothesisGenerator(C) do
12. if N(h) ≺ τ then h ← Revise(h) or discard
Output: Review, accepted hypotheses H

International Journal of Advanced Research