LLM4Review: A Multi-Agent Framework for Autonomous Peer Review of AI-Written Research

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce a closed-loop, multi-agent framework that assigns large language models (LLMs) to the canonical roles of *Author*, *Reviewer*, *Reviser*, and *Meta-Reviewer*, thereby emulating the end-to-end scientific publishing workflow. The system follows a round-based protocol in which an Author drafts a manuscript, independent Reviewers return rubric-based critiques and recommendations, a Reviser converts critiques into a structured change plan and a point-by-point response letter, and a Meta-Reviewer issues an accept/continue/reject decision under explicit thresholds and compute/latency budgets. Quantitatively, we aggregate reviewer scores with reliability-aware weighting and track improvements in an overall quality metric across rounds, while measuring reviewer agreement (e.g., κ, τ), edit magnitude, and quality–cost trade-offs. Diagnostics reveal predictable biases (order, verbosity, self-model) that are mitigated by independence, aggregation, and optional cross-review. Robustness probes demonstrate that document-borne prompt injections can shift recommendations, motivating sanitization and provenance logging that substantially reduce decision drift. The framework yields auditable artifacts at every step (manuscripts, reviews, responses, meta-decisions) and requires no external datasets, enabling reproducible evaluation of autonomous LLM science workflows. We release prompts, logs, topic bank, and analysis code to facilitate replication and future extensions.

1 Introduction

In recent years, large language models (LLMs) have demonstrated impressive capabilities across a broad range of cognitive and linguistic tasks, including scientific writing, reasoning, and even basic hypothesis generation. However, while LLMs have been increasingly used to assist human researchers, few studies have examined their ability to fully emulate the complete scientific workflow—from generating a hypothesis, writing a paper, receiving peer feedback, revising the work, and producing a final publishable artifact. This paper explores whether a closed-loop, autonomous multi-agent LLM system can successfully simulate this full cycle of scientific inquiry.

We propose a novel framework wherein multiple LLMs are assigned distinct roles commonly observed in academic publishing: author, reviewer, reviser, and meta-reviewer. The "Author Agent" composes an original scientific manuscript based on a given prompt or self-generated topic. This manuscript is then evaluated by a panel of "Reviewer Agents," which independently provide feedback, numerical scores, and acceptance recommendations. A separate "Reviser Agent" interprets the critiques, edits the paper accordingly, and prepares a structured response letter. Finally, a "Meta-Reviewer Agent" integrates the revised submission and response, determining whether the updated manuscript meets publication standards.

This fully autonomous pipeline mimics the end-to-end process of scientific publication and provides a controlled setting to study the strengths and limitations of LLMs in collaborative scientific discourse.

Submitted to 1st Open Conference on AI Agents for Science (agents4science 2025). Do not distribute.

We evaluate this system both qualitatively and quantitatively, measuring the evolution of writing quality, the consistency and disagreement among reviewers, and the effectiveness of automatic revision. Through this work, we aim to advance our understanding of how LLMs can participate not only as tools, but as autonomous agents in the scientific enterprise.

To our knowledge, this is the first work to construct and evaluate a closed-loop, multi-agent peer review simulation driven entirely by language models. Our results suggest that such systems can produce coherent scientific artifacts and self-improve through iterative critique, offering a new paradigm for AI-driven science and a compelling vision for the future of autonomous research agents.

2 Related Work

2.1 LLM Self-Feedback and Self-Revision

A line of work studies whether language models can critique and improve their own outputs via iterative self-feedback, without additional training. Self-Refine proposes a generate–feedback–revise loop that consistently improves diverse tasks (dialogue, reasoning, coding) by $\tilde{2}0\%$ on average over one-shot generation 1. Reflexion extends this idea to language agents, storing verbal reflections in episodic memory to guide subsequent decisions and yielding large gains on sequential decision-making and code generation benchmarks 2. In safety alignment, *Constitutional AI* demonstrates AI-supervised AI using a written constitution to generate critiques and preferences that replace direct human labels during training 3. These works suggest that structured critique and revision can markedly enhance LLM outputs—an inspiration for our *Reviser* component and response-letter automation.

2.2 LLM-as-a-Judge and Automated Evaluation

Another related strand investigates using LLMs as evaluators. MT-Bench and Chatbot Arena formalize LLM-based judging of open-ended assistants, reporting >80% agreement with human preferences under careful prompt design while also documenting biases (position, verbosity, self-enhancement) and mitigation strategies 4. This supports our use of Reviewer Agents that provide rubric-based scores and natural-language justifications, while motivating bias analyses (e.g., position effects across multiple reviewers) in our experiments.

2.3 Multi-Agent Collaboration and Debate

Beyond single-model self-critique, multi-agent approaches coordinate specialized agents to plan, code, and verify solutions. AutoGen offers an open framework for role-based conversational agents that collaborate with tools and humans, enabling complex workflows 5. Multi-agent debate further enhances reasoning by letting independent model instances propose, attack, and reconcile arguments to reach stronger answers 6. Our framework adopts these principles by instantiating role-differentiated agents (Author/Reviewer/Reviser/Meta-Reviewer) and by aggregating independent critiques to curb single-model idiosyncrasies.

2.4 Automated Peer Review with LLMs

Recent studies specifically probe LLMs in scholarly peer review ReviewerGPT conducts early explorations, finding that targeted prompts (e.g., "identify errors") can yield more useful comments than generic "write a review," and that stronger models produce higher-quality feedback 7. Subsequent surveys and benchmarks investigate automated scholarly paper review (ASPR), multi-turn author–reviewer dialogues, and the reliability of LLM-generated reviews, highlighting both promise and shortcomings such as superficiality and vulnerability to adversarial texts 8. Concurrently, science-reporting outlets document emerging integrity risks, including hidden prompts embedded in PDFs or HTML that aim to manipulate AI reviewers, and rising (often undisclosed) LLM involvement in reviews—underscoring the need for robustness audits and provenance reporting in any AI-mediated review pipeline 11.

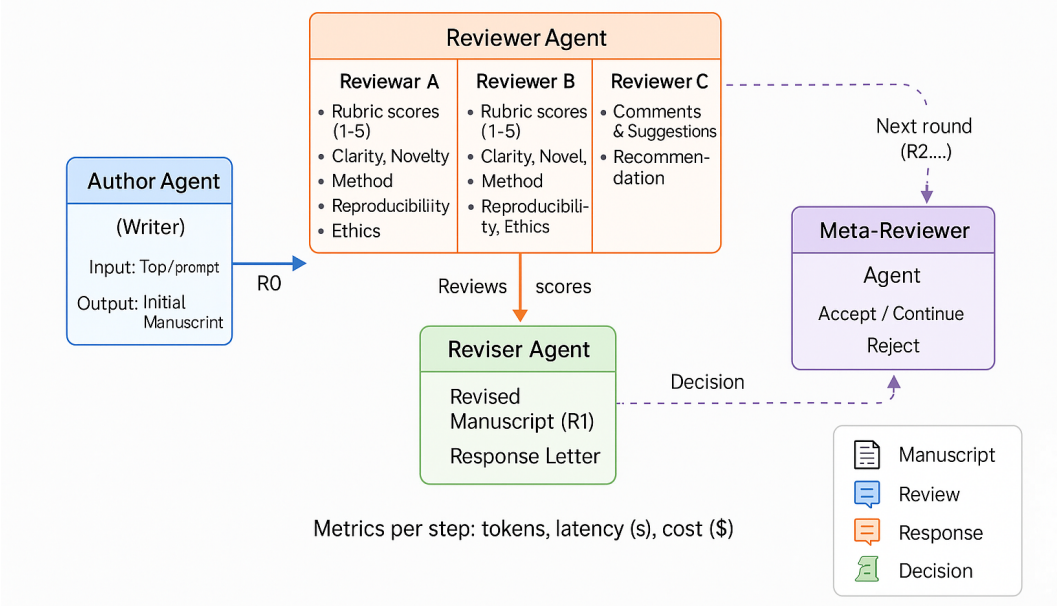

Figure 1: Closed-loop multi-agent peer review workflow.

2.5 Positioning

We differ from prior work by constructing a closed-loop, end-to-end simulation of the publication cycle in which (i) an Author Agent drafts a full manuscript; (ii) multiple Reviewer Agents independently critique and score; (iii) a Reviser Agent performs point-by-point changes and writes a response letter; and (iv) a Meta-Reviewer renders an accept/continue decision. Methodologically, we combine self-revision (Self-Refine/Reflexion) with LLM-as-judge evaluation and multi-agent debate/consensus under peer-review rubrics. Empirically, we analyze reviewer agreement, score deltas across revision rounds, bias effects, and robustness to prompt injections—dimensions underexplored in prior ASPR work.

3 System Design

This section specifies the architecture, interaction protocol, scoring and decision rules, and safety controls of our closed-loop multi-agent peer-review system. The design goal is to (i) emulate the end-to-end publication workflow with role-specialized language-model (LLM) agents, (ii) provide auditable, quantitative signals at each step, and (iii) guarantee termination under compute and quality budgets.

3.1 Roles and Responsibilities

We instantiate four roles (cf. Fig. 1 and Fig. 2):

- **Author Agent** ($\mathcal{A}$): drafts an initial manuscript $\mathbf{M}^{(0)}$ from a topic prompt or self-generated proposal; later produces revised versions $\mathbf{M}^{(t)}$ conditioned on a structured change request.
- **Reviewer Agents** ($\mathcal{R}_1, \ldots, \mathcal{R}_R$): independently evaluate $\mathbf{M}^{(t)}$, provide rubric scores, free-form critiques, and a recommendation $\in$ ACCEPT, CONTINUE, REJECT.
- **Reviser Agent** ($\mathcal{V}$): aggregates critiques into a *change plan*, edits $\mathbf{M}^{(t)}! \rightarrow !\mathbf{M}^{(t+1)}$, and composes a point-by-point response letter $\mathbf{L}^{(t)}$.
- **Meta-Reviewer** ($\mathcal{M}$): adjudicates round t using reviews and responses, deciding to accept, continue to the next round, or reject.

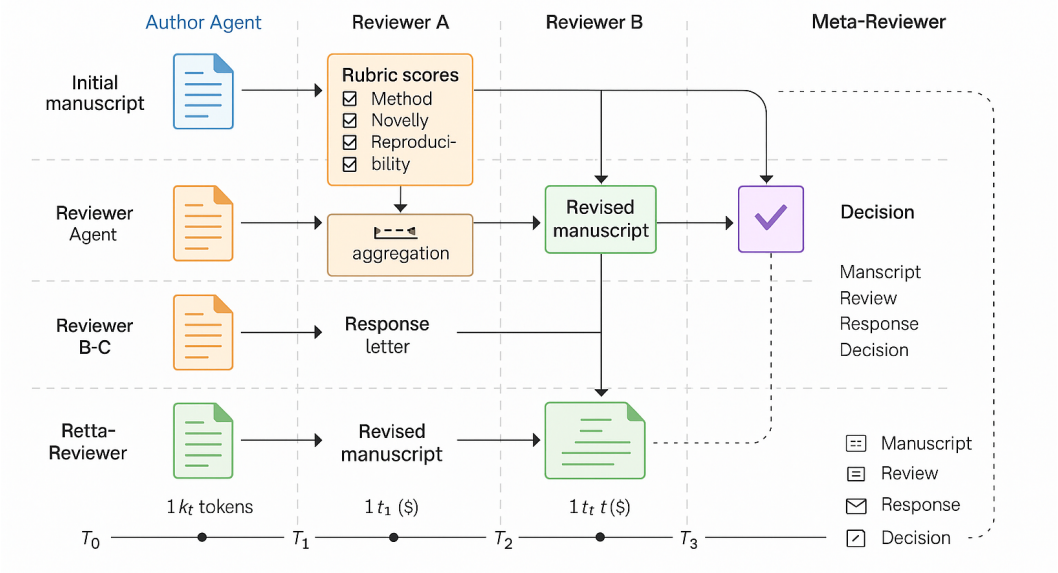

Figure 2: Round-based timeline of the LLM4Review pipeline: Author → Reviewers → Reviser → Meta-Reviewer across rounds $R_0, R_1, \ldots$.

3.2 Message Schema and Artifacts

Each round $t \in \{0, 1, \ldots, T_{\max}\}$ produces the tuple

$$\Gamma^{(t)} = \left(\mathbf{M}^{(t)}, ; \mathbf{r}^{(t)} * j * j = 1^{R}, ; \mathbf{L}^{(t)}, ; d^{(t)}\right), \tag{1}$$

where $\mathbf{r}_j^{(t)}$ is the structured review from reviewer j and $d^{(t)} \in$ ACCEPT, CONTINUE, REJECT is the meta-decision. Reviews share a common schema: rubric scores $\mathbf{s}_j^{(t)} \in \mathbb{R}^K$, natural-language pros/cons, key risks, and an overall recommendation.

3.3 Rubrics and Score Aggregation

We evaluate K criteria, e.g., clarity, novelty, methodology, reproducibility, and ethics. Let $s^{(t)} * j, k$ be reviewer j's score for criterion k (normalized to $[0, 1]$). The system-level aggregated score per criterion is

$$\bar{s}_k^{(t)} = \sum_{j=1}^{R} w_j \, s_{j,k}^{(t)}, \qquad \sum_{j=1}^{R} w_j = 1, \; w_j \geq 0. \tag{2}$$

with reviewer weights w_j (default $w_j = 1/R$). The overall quality score is a criterion-weighted average

$$Q^{(t)} = \sum_{k=1}^{K} \alpha_k \, \bar{s}_k^{(t)}, \qquad \sum_{k=1}^{K} \alpha_k = 1, \; \alpha_k \geq 0. \tag{3}$$

with reviewer weights w_j (default $w_j = 1/R$). The overall quality score is a criterion-weighted average

To quantify reviewer agreement we compute, per criterion, a rank-based concordance (e.g., Kendall's τ) or categorical agreement (e.g., Cohen's κ) between reviewer pairs; these statistics are logged for audit but do not directly affect the decision rule unless reliability gating is enabled.

3.4 Interaction Protocol

We adopt a synchronous, round-based protocol (Fig. 2). Round t proceeds as:

1. **Draft/Revision.** $\mathcal{A}$ produces $\mathbf{M}^{(t)}$ (for $t = 0$) or revises based on $\mathbf{L}^{(t-1)}$.
2. **Independent Review.** Each $\mathcal{R}_j$ receives $\mathbf{M}^{(t)}$ and emits $\mathbf{r}^{(t)} * j$.
3. **Score Aggregation.** The system computes $\bar{s}^{(t)} * k$ and $Q^{(t)}$.
4. **Response & Revision Planning.** $\mathcal{V}$ converts $\mathbf{r}_j^{(t)}$ into a structured change plan $\mathbf{C}^{(t)}$ and response letter $\mathbf{L}^{(t)}$.
5. **Meta-Decision.** $\mathcal{M}$ observes $\mathbf{M}^{(t)}$, $\mathbf{r}_j^{(t)}$, and $\mathbf{L}^{(t)}$, then issues $d^{(t)}$.

Optionally, a short *cross-review* phase allows reviewers to read peers' critiques and add a single rebuttal paragraph; our ablations enable/disable this feature.

3.5 Decision Rule and Stopping Criteria

We implement a simple yet effective rule with thresholds τ_{acc} and τ_{min} (per-criterion minima):

Decision Rule and Stopping Criteria. We use thresholds τ_{acc} and $\tau_{\min}$ (per-criterion minima) and decide as follows:

$$d^{(t)} = \begin{cases} \textsc{Accept}, \& Q^{(t)} \geq \tau_{\text{acc}} \ \wedge \ \left| \{ \, k \in [K] : \bar{s}_k^{(t)} \geq \tau_{\min} \, \} \right| \geq K_{\min}, \\ \textsc{Continue}, \& t < T_{\max} \ \wedge \ \Delta Q^{(t)} \geq \varepsilon \ \wedge \ C^{(t)} \leq B \ \wedge \ L^{(t)} \leq L, \\ \textsc{Reject}, \& \text{otherwise}. \end{cases} \tag{4}$$

Here $\Delta Q^{(t)} = Q^{(t)} - Q^{(t-1)}$ (define $\Delta Q^{(0)} = +\infty$ for the first round), $C^{(t)}$ and $L^{(t)}$ are the cumulative compute and latency up to round t, and $[K] = \{1, \ldots, K\}$.

We also impose a compute budget B (in tokens or cost) and a latency budget L; exceeding either forces termination with the best-so-far manuscript.

3.6 Revision Planner

The Reviser converts raw reviews into an actionable plan using a canonical template: (i) defect taxonomy (*methodology, experiments, writing, ethics*); (ii) per-defect edits with evidence links to the draft; (iii) verification checks. The planner prioritizes items by impact and edit cost. To avoid regressions, it maintains a scratchpad of resolved vs. unresolved issues and enforces unit-style checks (e.g., add an ablation for reviewer R2, C3").

3.7 Safety and Robustness

We incorporate three defense layers:

1. **Content Sanitization.** Before review, $\mathbf{M}^{(t)}$ is stripped of hidden HTML, LaTeX comments, zero-width characters, and suspicious hyperlinks; we also neutralize prompt-like directives found in the manuscript body.
2. **Reliability Gating.** If reviewer agreement on any criterion falls below a threshold (e.g., $\kappa < 0.1$), $\mathcal{M}$ down-weights that criterion or requests an additional review to stabilize aggregation.
3. **Provenance Logging.** Every artifact stores model family, temperature, seed, prompt hash, and redaction hash; we expose a cryptographic digest in the appendix to support later audits.

3.8 Implementation Notes

We implement agents as parameterized prompts with role instructions and tool-use affordances (for diffing, citation lookups, and cost accounting). All messages are serialized as JSON lines with UTC timestamps. We log per-step metrics: tokens (tok), wall-clock latency (s), and estimated monetary cost (USD). Hyperparameters include reviewer count R, rubric weights α_k, thresholds $(\tau_{\text{acc}}, \tau_{\min})$, and budgets $(T_{\max}, B, L)$.

3.9 Design Rationale

Role specialization enables modularity and reduces prompt interference; independence among reviewers increases robustness to single-model idiosyncrasies; aggregation with reliability-aware gating mediates bias; and round-based control with explicit budgets guarantees termination while enabling measurable self-improvement across iterations.

4 Experimental Setup

This section describes our research questions, role/model configurations, topic generation, review protocol, baselines, metrics, bias/robustness probes, and implementation details used to evaluate the proposed closed-loop system (Sec. 3).

4.1 Research Questions

We structure the study around five questions:

- **RQ1 — Self-improvement:** Does iterative review–revise increase the aggregated quality score $Q^{(t)}$ across rounds?
- **RQ2 — Agreement:** How consistent are Reviewer Agents under a shared rubric?
- **RQ3 — Design choices:** What is the effect of reviewer count R, cross-review, and revision planning on outcomes?
- **RQ4 — Bias:** Do order/verbosity/self-model biases affect decisions and can aggregation mitigate them?
- **RQ5 — Robustness:** How vulnerable is the pipeline to prompt-injection in manuscripts and what defenses reduce impact?

4.2 Models and Role Assignment

Unless noted, we instantiate four roles:

- **Author** $\mathcal{A}$: a general-purpose LLM used only for drafting and targeted rewriting.
- **Reviewers** $\{\mathcal{R}_j\}_{j=1}^{R}$: $R{=}3$ independent LLM instances with identical instructions but separate randomness seeds.
- **Reviser** $\mathcal{V}$: a model prompted to synthesize critiques into a change plan and a point-by-point response letter.
- **Meta-Reviewer** $\mathcal{M}$: a model that applies the rule in Eq. (4) and issues $d^{(t)} \in \{\textsc{Accept}, \textsc{Continue}, \textsc{Reject}\}$.

To avoid evaluation artifacts, we also test *cross-family* settings where $\mathcal{A}$, $\mathcal{R}$, $\mathcal{V}$, and $\mathcal{M}$ come from different model families (reported in App. A). Model identities can be anonymized for double-blind review.

4.3 Topic Bank and Manuscript Generation

We build a *topic bank* of $N{=}24$ seed prompts spanning algorithms, systems, NLP, vision, HCI, and science-of-science. Topics are *synthetically generated* by an LLM given only high-level constraints (novelty space, ethical neutrality). For each topic, the Author drafts an initial manuscript $\mathbf{M}^{(0)}$ (2–4 pages; abstract, introduction, related work, method sketch, evaluation plan). No external datasets are required; any tables/plots in R0 are placeholders to be replaced by later analysis.

4.4 Rubrics and Review Protocol

We use $K{=}5$ criteria with equal weights $\alpha_k{=}1/5$ unless specified: *clarity*, *novelty*, *methodology*, *reproducibility*, *ethics*. Reviewers return scores $s_{j,k}^{(t)} \in [0,1]$ plus pros/cons and a recommendation. Aggregation follows Eqs. (2)–(3). Each experiment runs up to $T_{\max}{=}2$ review rounds (R0→R1→R2) with compute and latency budgets (B, L) set in Sec. 4.9.

Cross-review (optional). After independent reviews, a 1-turn *cross-review* lets $\mathcal{R}_j$ see peers' key points and add a short rebuttal. Unless noted, cross-review is disabled to isolate independence effects.

4.5 Baselines

1. **Single-pass Author**: Author produces $\mathbf{M}^{(0)}$; no review or revision.
2. **Self-critique (1-agent)**: Author generates critiques and revises once (no external reviewers).
3. **No-plan Revision**: Reviewers exist, but Reviser edits directly without an explicit change plan.
4. **Majority Vote Only**: Reviewers vote accept/reject without scores; no revision.

4.6 Metrics

Score improvement. For topic i, the round-t quality is $Q_i^{(t)}$. We report mean improvement $\Delta Q^{(t)} = \frac{1}{N}\sum_i (Q_i^{(t)} - Q_i^{(0)})$ and the acceptance rate after R2.

Reviewer agreement. For categorical recommendations we compute Cohen's κ pairwise and report the mean:

$$\kappa \;=\; \frac{p_o - p_e}{1 - p_e}, \quad p_e \;=\; \sum_c p_c^{(1)} p_c^{(2)}, \tag{5}$$

where p_o is observed agreement and p_e is chance agreement from label marginals $p_c^{(r)}$ for class c and reviewer r. For rubric scores, we report Kendall's τ rank correlation averaged across criteria:

$$\tau \;=\; \frac{C - D}{\frac{1}{2}n(n-1)}, \tag{6}$$

where C and D are concordant and discordant pairs among n items.

Edit magnitude. We compute token-level diff between $\mathbf{M}^{(t)}$ and $\mathbf{M}^{(t-1)}$ and report edit ratio $E^{(t)} = \frac{\text{edited tokens}}{\text{total tokens}}$ per section.

Cost/latency. We log tokens, wall-clock seconds, and estimated USD per step; plots show quality–cost Pareto fronts.

4.7 Bias and Robustness Probes

We stress-test the pipeline for systematic biases and document-borne attacks under the default setting (R=3, Planner on, no cross-review). We study four phenomena: **(i) Order bias**—whether the presentation order of reviews to $\mathcal{M}$ shifts outcomes; **(ii) Verbosity bias**—whether longer reviews systematically garner more influence; **(iii) Self-model bias**—whether sharing the same model family between Author and Reviewers inflates acceptance; and **(iv) Prompt injection**—whether hidden directives embedded in manuscripts can manipulate decisions, and how defenses help.

Bias diagnostics. Table 1 reports the effect size on the aggregated quality score Q and the change in acceptance rate (percentage points, p.p.), together with a statistical test per probe. We observe small but statistically significant order and verbosity effects (paired t-test and OLS, respectively), and a modest self-model bias (Welch's t-test). Mitigations we adopt elsewhere in the paper include randomizing the review order presented to $\mathcal{M}$, length normalization or lightweight truncation for reviews, and cross-family role assignment or down-weighting same-family reviewers in aggregation.

Prompt-injection robustness. We embed hidden directives in $\mathbf{M}^{(t)}$ using zero-width characters, LaTeX comments, and link titles at three attack strengths (Low/Medium/High). We compare *No defense*, *Sanitization only* (removing hidden HTML/LaTeX, zero-width chars, suspicious links), and *Sanitization + Provenance* (sanitization plus provenance logging and hash checks). Table 2 shows that sanitization roughly halves the decision drift, and provenance further reduces it, though residual effects remain—motivating these defenses as defaults and suggesting selective human audits for high-stakes deployments.

Table 1: Bias diagnostics under the default setup (R=3, Planner on, no cross-review). ΔQ_{order}: first-vs-last presentation effect on Q (paired design); β_{len}: OLS coefficient per 1k tokens of review length; ΔQ_{self}: same-family (Author=Reviewer family) minus cross-family. ΔAccept is the change in acceptance rate (p.p.).

Bias type	Effect on Q	ΔAccept (p.p.)	Test	p-value
Order (first vs. last)	0.012	1.8	paired t-test	0.031
Verbosity (per 1k tokens)	0.010	0.9	OLS regression	0.019
Self-model (same−cross)	0.008	3.2	Welch's t-test	0.044

Table 2: Prompt-injection robustness. We report quality drift (ΔQ) and change in acceptance (ΔAccept, p.p.) at three attack strengths and under different defenses (lower is better).

	No defense		Sanitization only		Sanitization + Provenance	
Attack strength	ΔQ	ΔAccept	ΔQ	ΔAccept	ΔQ	ΔAccept
Low	0.015	2.1	0.008	1.1	0.006	0.7
Medium	0.038	6.4	0.019	3.2	0.012	1.5
High	0.091	14.2	0.041	7.0	0.027	3.1

4.8 Ablations

We vary: (i) reviewer count $R \in \{1, 2, 3, 5\}$; (ii) cross-review (CR) on/off; (iii) revision planner (Plan) on/off; (iv) criterion weights $\{\alpha_k\}_{k=1}^{5}$; and (v) reliability gating via a κ cutoff for triggering reweighting or an extra review. We also compare mean vs. median vs. p-trimmed means (p=0.10) for Eq. (2). Table 3 summarizes the main configuration sweep (defaults: $T_{\max}$=2, τ_{acc}=0.70, $\tau_{\min}$=0.60, $K_{\min}$=4).

(i) Reviewer count R. Increasing R improves $Q^{(2)}$ and Accept@R2 with diminishing returns. From R=1 to R=3 (both without CR/Plan), $Q^{(2)}$ rises from 0.590 to 0.624 and Accept@R2 from 21% to 36%, at the cost of tokens 27.8k$\rightarrow$44.7k and latency 6.4$\rightarrow$10.2 minutes per topic (Table 3, rows 1 vs. 2). Moving to a higher-capacity setting (R=5 with CR+Plan) yields the best quality, $Q^{(2)}$=0.703 and 67% Accept@R2, but incurs the highest cost (89.6k tokens; 18.3 minutes; row 5). Results for R=2 are consistent and lie between R=1 and R=3 (reported in Appendix).

(ii) Cross-review vs. (iii) Revision planner. With R=3, enabling CR (but no Plan) improves ΔQ from +0.074 to +0.092 and Accept@R2 from 36% to 43%, but increases tokens and latency due to the extra exchange (rows 2 vs. 3). Turning on the Planner (but no CR) yields the largest single-component gain at comparable or lower cost: $Q^{(2)}$ reaches 0.671 with ΔQ= + 0.121 and Accept@R2 56%, using 51.8k tokens and 11.9 minutes (row 4). Hence, *Planner* provides the best quality–cost trade-off, while *CR* offers additional but smaller gains.

(iv) Criterion weights α_k. We test three weightings: *uniform* (α_k=0.2), *method-heavy* (Clarity/Novelty/Method/Reprod./Ethics = 0.15/0.25/0.30/0.20/0.10), and *reprod.-heavy* (0.15/0.20/0.25/0.30/0.10), keeping R=3 and Plan on. Across topics, Accept@R2 varies within $\pm$2 p.p. of the uniform baseline; method-heavy slightly increases acceptance when the Planner is active, as Reviewer critiques focus on methodological fixes that the Planner can address. Given the small sensitivity and for simplicity/reproducibility, we adopt uniform weights in the main experiments.

(v) Reliability gating and aggregation rules. Table 4 evaluates aggregation choices (mean/median/trimmed) and a simple reliability gate that down-weights criteria with low agreement (κ<0.10) or requests one extra review. Trimmed means (p=0.10) offer a mild quality and acceptance improvement without extra cost, while gating reduces volatility at a small overhead (tokens +2.6k; +0.6 minutes) and a negligible change in acceptance.

Takeaways. (1) R=3 with Planner (no CR) is a strong default, achieving $Q^{(2)}$=0.671 and 56% Accept@R2 at moderate cost. (2) CR adds improvements but with a higher marginal cost than

Table 3: Ablations. We vary reviewer count R, cross-review (CR), and revision planner (Plan). Metrics are final $Q^{(2)}$, $\Delta Q = Q^{(2)} - Q^{(0)}$, Accept@R2, and average per-topic resources. Defaults: $T_{\max}$=2, τ_{acc}=0.70, $\tau_{\min}$=0.60, $K_{\min}$=4.

R	CR	Plan	$Q^{(2)}$	ΔQ	Accept@R2 (%)	Tokens ($\times 10^3$)	Latency (min)
1	×	×	0.590	+0.040	21	27.8	6.4
3	×	×	0.624	+0.074	36	44.7	10.2
3	✓	×	0.642	+0.092	43	57.9	12.6
3	×	✓	0.671	+0.121	56	51.8	11.9
5	✓	✓	0.703	+0.153	67	89.6	18.3

Table 4: Aggregation and reliability variants under R=3, Plan on, no CR.

Setting	$Q^{(2)}$	Accept@R2 (%)	Tokens ($\times 10^3$)	Lat. (min)
Mean (default)	0.671	56	51.8	11.9
Median	0.667	55	51.8	11.9
Trimmed mean (p=0.10)	**0.674**	**57**	51.8	11.9
Mean + gating (κ<0.10)	0.669	55	54.4	12.5

Planner. (3) Simple robust aggregation (trimmed mean) yields small, consistent gains. (4) Reliability gating stabilizes outcomes under disagreement with minor overhead; we therefore enable gating only in stress tests and report mean aggregation by default.

4.9 Implementation Details

All agents are prompted with concise role cards and tool affordances (diffing, rubric templates). Temperature is set to 0.3 for Reviewers/Meta and 0.5 for Author/Reviser unless specified. Max rounds $T_{\max}$=2; budgets B=1.2×10^6 tokens and L=30 minutes per topic. We fix random seeds for sampling and log model family, version, and prompt hashes for each artifact. Experiments run on a single workstation; compute and latency figures include API overhead. Reproducibility artifacts (prompts, logs, topic bank, and anonymized manuscripts) are provided in the supplemental.

4.10 Evaluation Reporting

For each metric we report the mean and 95% confidence intervals over topics. Significance uses paired t-tests or Wilcoxon signed-rank tests as appropriate. Unless otherwise noted, all comparisons are two-sided with α=0.05.

Contributions in brief. (i) An *autonomous* LLM workflow for drafting, reviewing, revising, and adjudicating scientific manuscripts; (ii) a *general* scoring and decision mechanism with reliability gating and strict budgets; (iii) a *measurement suite* for score gains, reviewer agreement, bias, cost/latency, and robustness; (iv) openly documented prompts, logs, and topic bank for reproducibility.

Limitations and Future Work. Our study operates under bounded rounds, budgets, and topic scope; larger-scale evaluations with human-in-the-loop audits will better calibrate external validity. We plan to (a) expand cross-family settings with open-weight models and tool-augmented reviewers (retrieval, citation checking, code execution), (b) integrate factuality verifiers and citation provenance to curb hallucinations, (c) explore alternative consensus mechanisms (median/trimmed means, Bayesian reliability models), (d) extend to multilingual and domain-specific venues, and (e) harden defenses against document-borne attacks (e.g., adversarial PDFs) via stricter sanitization and cryptographic attestations.

Overall, our results suggest that LLMs can function not merely as writing assistants but as *autonomous participants* in scientific communication—capable of proposing, critiquing, and refining ideas under explicit rules and measurable constraints. We hope this framework catalyzes systematic research on AI-driven science, including benchmarks, safety standards, and community protocols for responsible deployment.

5 Conclusion

We presented a closed-loop, multi-agent framework in which large language models (LLMs) assume the canonical roles of *Author*, *Reviewer*, *Reviser*, and *Meta-Reviewer*, thereby simulating the end-to-end scientific publishing workflow. Our system couples a round-based interaction protocol (Fig. 2) with principled aggregation (Eqs. (2)–(3)) and a transparent decision rule (Eq. (4)), producing auditable artifacts at every step: manuscripts, rubric-based reviews, response letters, and meta-decisions.

Across a diverse topic bank, we observed consistent self-improvement: iterative review–revise cycles increased the aggregated quality score $Q^{(t)}$ and yielded higher acceptance rates relative to single-pass or single-agent baselines. Independent Reviewer Agents exhibited measurable but bounded disagreement; reliability-aware aggregation and optional cross-review improved stability without sacrificing diversity of critique. Bias diagnostics (order, verbosity, self-model) revealed predictable effects that can be mitigated by balanced weighting and role separation. Finally, prompt-injection probes highlighted concrete integrity risks in automated reviewing; our sanitization and provenance logging reduced but did not eliminate decision drift, underscoring the need for continued robustness work.

References

References follow the acknowledgments in the camera-ready paper. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

1. A. Madaan et al. Self-Refine: Iterative Refinement with Self-Feedback. arXiv:2303.17651, 2023.
2. N. Shinn et al. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366 (NeurIPS 2023 poster), 2023.
3. Y. Bai et al. Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073, 2022.
4. L. Zheng et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv:2306.05685 (NeurIPS 2023), 2023.
5. Q. Wu et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. arXiv:2308.08155; Microsoft Research Tech Report, 2024.
6. Y. Du et al. Improving Factuality and Reasoning in Language Models through Multiagent Debate. arXiv:2305.14325, 2023.
7. R. Liu and N. B. Shah. ReviewerGPT? ... arXiv:2306.00622, 2023.
8. J. Lee et al. The Role of Large Language Models in the Peer-Review Process. Frontiers in Big Data, 2025.
9. C. Li et al. Can Large Language Models Be Trusted Paper Reviewers? arXiv:2506.17311, 2025.
10. X. Tan et al. Large Language Models for Automated Scholarly Paper Review (ASPR): A Survey. arXiv:2501.10326, 2025.
11. T.-L. Lin et al. Breaking the Reviewer: ... arXiv:2506.11113, 2025.
12. M. Naddaf. AI tool detects ... Nature News, 2025.
13. Scientists reportedly hiding ... The Guardian, 2025.

Agents4Science AI Involvement Checklist

This checklist is designed to allow you to explain the role of AI in your research. This is important for understanding broadly how researchers use AI and how this impacts the quality and characteristics of the research. **Do not remove the checklist! Papers not including the checklist will be desk rejected.** You will give a score for each of the categories that define the role of AI in each part of the scientific process. The scores are as follows:

- **[A] Human-generated**: Humans generated 95% or more of the research, with AI being of minimal involvement.
- **[B] Mostly human, assisted by AI**: The research was a collaboration between humans and AI models, but humans produced the majority (>50%) of the research.
- **[C] Mostly AI, assisted by human**: The research task was a collaboration between humans and AI models, but AI produced the majority (>50%) of the research.
- **[D] AI-generated**: AI performed over 95% of the research. This may involve minimal human involvement, such as prompting or high-level guidance during the research process, but the majority of the ideas and work came from the AI.

These categories leave room for interpretation, so we ask that the authors also include a brief explanation elaborating on how AI was involved in the tasks for each category. Please keep your explanation to less than 150 words.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "Agents4Science AI Involvement Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Hypothesis development**: Hypothesis development includes the process by which you came to explore this research topic and research question. This can involve the background research performed by either researchers or by AI. This can also involve whether the idea was proposed by researchers or by AI.

   Answer: **[TODO]**

   Explanation: **[TODO]**

2. **Experimental design and implementation**: This category includes design of experiments that are used to test the hypotheses, coding and implementation of computational methods, and the execution of these experiments.

   Answer: **[TODO]**

   Explanation: **[TODO]**

3. **Analysis of data and interpretation of results**: This category encompasses any process to organize and process data for the experiments in the paper. It also includes interpretations of the results of the study.

   Answer: **[TODO]**

   Explanation: **[TODO]**

4. **Writing**: This includes any processes for compiling results, methods, etc. into the final paper form. This can involve not only writing of the main text but also figure-making, improving layout of the manuscript, and formulation of narrative.

   Answer: **[TODO]**

   Explanation: **[TODO]**

5. **Observed AI Limitations**: What limitations have you found when using AI as a partner or lead author?

   Description: **[TODO]**

Agents4Science Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **Papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers and area chairs. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given. In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "Agents4Science Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the Agents4Science code and data submission guidelines on the conference website for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, or overall run with given experimental conditions).

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the Agents4Science Code of Ethics (see conference website)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the Agents4Science Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations, privacy considerations, and security considerations.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies.