Experts First, Iteration Second: Auditable Self-Improvement for Scientific Peer-Review Agents

Lele Cao
CSPaper, Scholar7
Stockholm, Sweden
lele@scholar7.com

Xin Huang
CSPaper, Scholar7
Ballerup, Denmark
xin@scholar7.com

Lei You
CSPaper, Scholar7
Ballerup, Denmark
lei@scholar7.com

Abstract

Self-improving agents are often framed as recursive systems that discover their own improvement procedures. For high-stakes vertical NLP systems, however, the bottleneck is often not autonomy but the placement of expert knowledge: domain specialists can supply rubrics, failure modes, calibration examples, and deployment constraints that an open search loop would otherwise need to rediscover. We present SIRA (self-improving review agent), an expert-bootstrapped agent factory for scientific peer-review support. SIRA keeps the online reviewer and common execution harness fixed, while offline iterations edit only venue-specific artifacts: rubrics, metadata, prompts, templates, calibration rules, benchmark packs, and failure analyses. On a paper-review agent-creation task, SIRA achieves a mean best held-out decision-label accuracy of 0.941 over five runs, compared with 0.865 for a HyperAgents-style open editable-agent baseline under the same dataset split and metric; it also reaches its best candidate in roughly one third as many scored steps. The claim is bounded but sharp: in peer-review support, self-improvement can be strongest when experts shape the search space first and iteration is restricted to auditable, versioned factory artifacts.

1 Introduction

Self-improving agents are usually imagined as increasingly open-ended systems: the agent changes its behavior, changes the procedure that changes its behavior, and accumulates improvements through exploration (Schmidhuber, 2007; Zhang et al., 2025a, 2026). That vision is scientifically important. It is also not the only useful systems abstraction. For high-stakes vertical NLP systems, the decisive question is often not how much autonomy an agent can acquire, but how much domain structure should be supplied before iteration begins.

This paper studies that inversion: *experts first, iteration second*. In scientific peer-review support, domain experts already know many of the structures that an unconstrained search loop would need to rediscover: official rubrics, common review failures, score-calibration pathologies, venue-specific expectations, and cases where fluent feedback becomes misleading. The system challenge is therefore not to remove experts from the loop, but to convert their knowledge into an auditable improvement substrate.

Peer-review support makes this boundary problem concrete. Large language models can provide useful manuscript feedback, but they also produce unreliable criticism, shallow substantiation, and fragile score predictions (Liang et al., 2024; Liu and Shah, 2023; Zhou et al., 2024; Shin et al., 2025; Guo et al., 2023; Dycke et al., 2025; Thelwall and Yaghi, 2025; Bao et al., 2021; Zhang et al., 2025b). A review assistant may help authors find missing evidence, weak framing, unclear novelty, and benchmark gaps before submission; it must not silently learn from private submissions or present historical accept/reject imitation as scientific truth.

We study this question through **SIRA**, the **self-improving review-agent** factory used to create venue-specific agents for CSPR, a public computer-science paper feedback service (Cao et al., 2025). SIRA repeatedly regenerates, evaluates, repairs, versions, and archives fixed review agents. It does not deploy a self-modifying reviewer. The resulting design is intentionally asymmetric: expert-maintained factory skills and engineer-maintained utilities define what may change; venue-specific artifacts and training benchmark sidecars change offline; held-out validation packs and online service behavior remain fixed during a run. Figure 1 summarizes this boundary.

The core contribution is a systems claim and a budgeted-search lens. We model review-agent creation as finite-budget search over versioned agent bundles, where expert-maintained rubrics, calibration examples, and failure diagnostics shape the

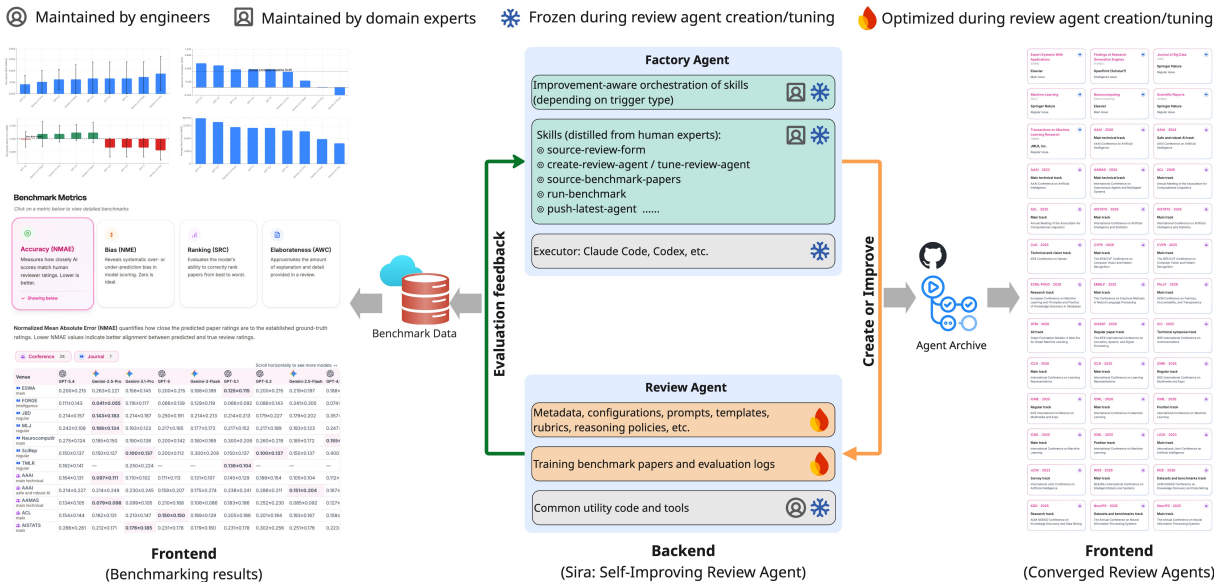


Figure 1: SIRA turns self-improvement into an offline, auditable agent-creation loop. Engineers maintain the serving harness and common utilities; domain experts maintain factory skills, review rubrics, calibration examples, and failure diagnostics. During service, the deployed review agent is fixed. During offline creation/finetune, the factory regenerates venue-specific artifacts, audits failures, evaluates candidates, and promotes only versioned agents.

proposal distribution before offline iteration begins. Under the same base model, 70/30 re-split paper-review dataset, candidate-scoring budget, evaluation harness, and held-out validation metric, SIRA outperforms a HyperAgents-style open editable-agent baseline (Zhang et al., 2026)in both peak validation accuracy and steps to best candidate. We do not claim that constrained factories dominate open-ended self-improvement in general. We claim that, for high-volume scientific review support, the factory boundary is a better abstraction because it treats domain expertise as search compression under finite evaluation budget.

2 Self-Improvement at Factory Boundary

SIRA is an agent-building system rather than a self-modifying task agent. For a venue v, a deployed review agent is a versioned bundle

$$a_v = (m_v, r_v, p_v, t_v, c_v, u), \tag{1}$$

where m_v is venue metadata, r_v official rubrics and calls for papers, p_v prompts and reasoning policies, t_v review templates, c_v calibration and score-mapping rules, and u common utility code inherited from the serving harness. A separate sidecar b_v contains training benchmark papers, expected labels, evaluation records, and failure tickets. The sidecar is used to create and tune agents; it is not deployed with online reviewers. A factory step is

$$(a_v^{(k+1)}, b_v^{(k+1)}) \leftarrow F_\phi(a_v^{(k)}, b_v^{(k)}, E(a_v^{(k)}; b_v^{(k)}), S_v, H_v), \tag{2}$$

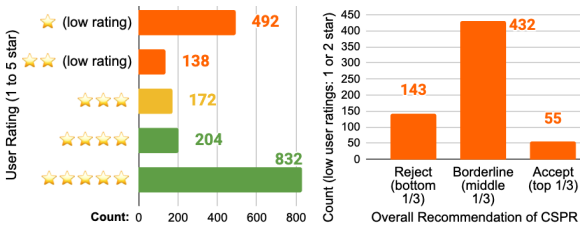
where F_ϕ is the factory controller, E is the evaluation harness, S_v is the expert-maintained skill library, and H_v is explicit human guidance. During a creation run, F_ϕ, E, the active version of S_v, and the common utilities are fixed for auditability. Between runs, engineers and domain experts may update those components through ordinary versioned maintenance. Within a run, improvement is restricted to venue artifacts and benchmark sidecars.

The allowed actions are deliberately prosaic: source official venue information, create or tune an agent, curate benchmark papers, run evaluation, inspect failures, and push a named version to the archive. The point is not to reduce adaptability, but to make the locus of change inspectable. Table 1 contrasts this factory boundary with an open editable-agent baseline, and Appendix A gives additional implementation details.

The same boundary defines the human roles. Engineers maintain the executor, deployment, logs, and shared utilities. Domain experts maintain factory skills, rubrics, calibration examples, and diagnostic templates. The factory uses those artifacts to generate candidate review agents, but the online reviewer remains frozen until a new version passes evaluation and is explicitly promoted. This resembles continuous integration, except the tested unit is an agent bundle containing prompts, retrieval behavior, score mappings, templates, and benchmark.

Table 1: Design contrast. SIRA gives up some open-endedness to satisfy production constraints that matter for unpublished scientific manuscripts.

Dimension	Open editable-agent baseline	SIRA factory regeneration	Practical consequence
Editable object	Task behavior and the procedure generating future task behavior may both change.	Factory controller, active skill library, and common utilities are frozen during a run; only venue artifacts and benchmark sidecars evolve.	Clearer ownership, auditability, and rollback.
Improvement site	Improvement may be coupled to an autonomous task loop.	Improvement is offline: regenerate, evaluate, version, redeploy.	Private manuscripts are processed by fixed agents.
Search bias	Broad exploration can discover mechanisms but can spend budget rediscovering domain structure.	Search begins with official rubrics, known review failures, calibrated examples, and diagnostics.	Faster time-to-value for narrow vertical products.
Promotion evidence	Feedback can become entangled with evolving meta-procedures.	Fixed validation packs, refreshed benchmark samples, rubric diagnostics, and human failure triage.	Promotion decisions are easier to replay and contest.
Risk profile	More expressive but harder to bound when self-modification mechanisms change.	Less general but designed for privacy, governance, and predictable service behavior.	Suitable for pre-review support, not automated peer review.



(a) All ratings (b) Low ratings

Figure 2: Post-review quality ratings from (Cao et al., 2026). Low-rated reviews are not used as online training data. They are converted, when allowed, into de-identified tickets, aggregate diagnostics, or benchmark-refresh candidates inspected before offline tuning.

3 Signals from a Deployed Review Service

Peer-review support systems need evaluation signals beyond historical decision labels. CSPR collects post-delivery user ratings of generated review reports and audits low-rated cases as operational failure signals when policy permits. Figure 2 shows one production snapshot: $n = 1{,}838$ post-review ratings, including 630 one- or two-star ratings. Low ratings concentrate heavily around borderline recommendations, which is precisely where score calibration, uncertainty communication, and explanation quality are hardest.

These ratings are intentionally not treated as reinforcement for a live agent. A private manuscript can reveal a system failure, but it should not silently alter the system that reviews the next manuscript. In SIRA, feedback becomes a ticket or aggregate diagnostic; domain experts decide whether it should update a rubric, template, calibration rule, benchmark sample, or failure test. This slow path is part of the method: it preserves the distinction between evidence about the system and automatic modification of the system. Appendix B discusses why borderline cases are useful as diagnostic signals.

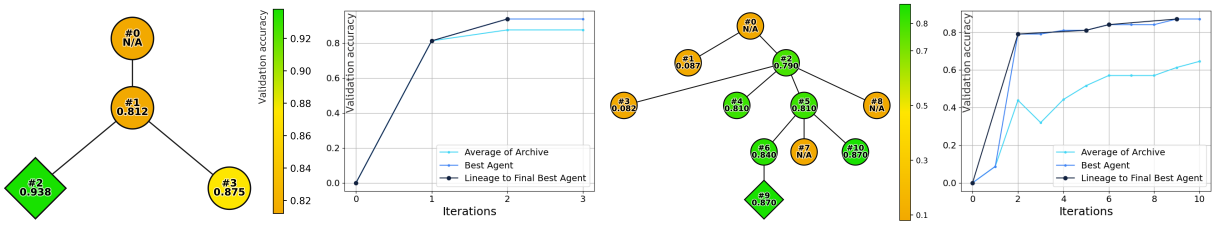
4 Comparative Study

We evaluate SIRA in the same paper-review domain studied by HyperAgents (Zhang et al., 2026). Each sample contains the full text of an AI research paper and a binary accept/reject label derived from real conference decisions. The label is a decision-side proxy: it measures agreement with historical outcomes, not the factual correctness, helpfulness, or epistemic quality of generated criticism. We re-split the available dataset into a fixed 70/30 train–validation partition, preserving the split and class balance across all runs.

The training split is available to the creation loop. SIRA creates, repairs, and evolves candidate ICLR-style review agents using training feedback, benchmark failures, and human-inspected observations. The validation split is held outside this evolution process and used only to score generated candidates and visualize the search trajectory. The baseline is a HyperAgents-style adaptation in which task behavior and meta-improvement behavior are represented as a jointly editable archive. We use "HyperAgents-style" because the comparison adapts the mutable task/meta-agent design to the review-agent creation task; it is not an official reproduction of every experimental choice in Zhang et al. (2026). Both settings use the same base model, dataset split, candidate-scoring budget, evaluation harness, and validation metric. The main difference is the mutability boundary.

Figure 3 visualizes representative search traces under this protocol, and Table 2 reports five-run summaries. Appendix C states the experimental scope and data-use boundary.

The results support a bounded practical claim: in this review-agent creation setting, the constrained factory reaches higher held-out decision-label accu-



(a) SIRA tree (b) SIRA progress (c) Open baseline tree (d) Open baseline progress

Figure 3: Representative self-improvement traces. Nodes are archive candidates; colors encode validation accuracy when available; N/A marks unscored nodes; diamonds mark the final best candidate. The constrained factory reaches a high-scoring agent with fewer scored candidates because the search space is shaped by venue rubrics, expert skills, and benchmark-aware failure analysis.

Table 2: Five-run validation summary on the re-split paper-review dataset. Higher is better except for steps.

Metric	SIRA	HyperAgents-style
Best validation accuracy	**0.941 ± 0.018**	0.865 ± 0.049
Final mean archive accuracy	**0.868 ± 0.023**	0.590 ± 0.072
Steps to best, range	**2–5**	6–15
Mutable meta-agent in run	No	Yes

"Best" is the highest held-out validation accuracy reached by any scored candidate. "Final mean" averages scored candidates in the final archive to expose archive stability rather than only peak performance.

racy on average, maintains a stronger final archive, and needs fewer scored steps to find its best candidate. A compact way to state the mechanism is budgeted expert-compressed search. Let D_{val} denote the validation dataset, and let B be the candidate-scoring budget. A creation method M, i.e. SIRA or the open editable-agent baseline, induces a possibly adaptive distribution Q_M over scored candidate traces $a_{1:B} = (a_1, \ldots, a_B)$, where each a_i is a candidate review-agent bundle for venue v. Let $s_v(a_i; D_{\text{val}})$ be the held-out validation score of candidate a_i. We summarize finite-budget search by the expected best held-out score within budget,

$$\mathcal{S}_B(M) = \mathbb{E}_{a_{1:B}\sim Q_M}\left[\max_{1\le i\le B} s_v(a_i; D_{\text{val}})\right],$$

and, when a target quality level τ is specified, by the probability of reaching that level within budget,

$$\Pr_{a_{1:B}\sim Q_M}\left[\max_{1\le i\le B} s_v(a_i; D_{\text{val}}) \ge \tau\right].$$

The expectation and probability are over the randomness of the creation run, including candidate generation, repair choices, and run-level variation. SIRA's hypothesis is that expert rubrics, calibration examples, and benchmark-aware failure analysis shift Q_M toward higher-density useful candidates before scoring begins. The open baseline is more expressive, but under a finite scoring budget it may spend evaluations rediscovering structure already supplied by domain experts.

5 Implications for Scientific NLP Agents

The budgeted-search view implies that self-improvement for scientific NLP agents is a mutability-boundary problem. SIRA makes the deployed reviewer stable, moves candidate improvement offline, and requires promotion through validation packs, diagnostic samples, rubric-level tests, and qualitative failure audits. This boundary is especially important for peer-review support: agents should provide evidence-linked, contestable critiques for authors, not acceptance decisions or silent online learning from private manuscripts. These commitments are compatible with broader work on grounding and retrieval (Lewis et al., 2020; Asai et al., 2023; Gao et al., 2023; Thorne et al., 2018), truthful and factual generation (Lin et al., 2022; Manakul et al., 2023; Min et al., 2023), documentation (Mitchell et al., 2019; Gebru et al., 2021), and agent evaluation (Liang et al., 2022; Chen et al., 2021; Zheng et al., 2023; Liu et al., 2023; Wang et al., 2024). They also respect the social nature of peer review, where assignment, arbitrariness, and inconsistency shape outcomes (Roos et al., 2011; Langford and Guzdial, 2015; Cortes and Lawrence, 2021; Shah, 2022; Kang et al., 2018; Li et al., 2020). Appendix D distills these commitments into a responsible-use checklist.

6 Conclusion

SIRA reframes self-improvement as experts-first offline iteration over auditable artifacts, rather than online self-modification by deployed reviewers. In peer-review support, this boundary improves search efficiency, early performance, and auditability.

Limitations

The evidence is intentionally narrow: one review-agent creation setting, one binary decision-label proxy, one validation split, and short search traces. Agreement with historical accept/reject decisions does not measure review quality itself. A high decision-label score can still coincide with unhelpful criticism, factual errors, weak substantiation, unfair topic treatment, institutional bias, regional bias, poor robustness to prompt injection, or miscalibration under distribution shift. The HyperAgents-style comparison is a controlled adaptation designed to test a mutability-boundary hypothesis, not a final benchmark of HyperAgents. Future work should report held-out test performance across multiple venues, ablate expert skills, measure token cost and latency, run human preference and author-outcome studies, and audit privacy and leakage failures.

Ethical Considerations

SIRA and CSPR are designed for pre-submission support and computational research assessment, not for replacing program committees or reviewers. Review agents should not be used as authoritative acceptance predictors, should not silently learn from private submissions, and should not be presented as impartial judges of scientific value. Their intended role is earlier and narrower: help authors find missing evidence, weak claims, unclear contributions, benchmark gaps, and rubric mismatches before submission. Private manuscripts and user ratings can expose failures, but any use of such signals should be de-identified, policy-permitted, and mediated through offline evaluation rather than online self-modification.

Use of AI Assistants. LLM-based writing assistants (mainly GPT series models) were used in a limited manner for grammar correction and light editorial refinement. The authors verified and finalized all technical content, experimental descriptions, claims, and conclusions.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

Peng Bao, Weihui Hong, and Xuanya Li. 2021. Predicting paper acceptance via interpretable decision sets. In *Companion Proceedings of the Web Conference 2021*, pages 461–467.

Lele Cao, Lei You, and R&D Team. 2025. CSPaper review: Fast, rubric-faithful conference feedback. In *Proceedings of the 18th International Natural Language Generation Conference: System Demonstrations*, pages 3–7, Hanoi, Vietnam. Association for Computational Linguistics.

Lele Cao, Lei You, Kai Xie, Weiping Ding, Yong Du, Sven Salmonsson, Yumin Zhou, and Vilhelm von Ehrenheim. 2026. Adopt machine-human collaboration peer-review through computational research assessment. *OpenPrint:20260212.0002v1*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Corinna Cortes and Neil D. Lawrence. 2021. Inconsistency in conference peer review: Revisiting the 2014 neurips experiment. In *NeurIPS 2021 Workshop on Meta-Learning*.

Nils Dycke, Matej Zečević, Ilia Kuznetsov, Beatrix Suess, Kristian Kersting, and Iryna Gurevych. 2025. STRICTA: Structured reasoning in critical text assessment for peer review and beyond. *Preprint*, arXiv:2409.05367. Accepted at ACL 2025.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Yanzhu Guo, Guokan Shang, Virgile Rennard, Michalis Vazirgiannis, and Chloé Clavel. 2023. Automatic analysis of substantiation in scientific peer reviews. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10198–10216.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications. In *Proceedings of NAACL-HLT*, pages 1647–1661.

John Langford and Mark Guzdial. 2015. The arbitrariness of reviews, and advice for school administrators. Communications of the ACM Blog.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Siqing Li, Wayne Xin Zhou, and Xiaodan Zhu. 2020. A multi-task peer-review score prediction model. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 121–126.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *Preprint*, arXiv:2211.09110.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, and 1 others. 2024. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8).

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 3214–3252.

Ryan Liu and Nihar B. Shah. 2023. Reviewergpt? an exploratory study on using large language models for paper reviewing. *Preprint*, arXiv:2306.00622.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *Preprint*, arXiv:2303.08896.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229.

Mark Roos, Jorg Rothe, and Bjorn Scheuermann. 2011. Reviewer assignment problem: A systematic review of the literature. *Journal of Artificial Intelligence Research*, 42:523–545.

Jurgen Schmidhuber. 2007. Godel machines: Fully self-referential optimal universal self-improvers. *Artificial General Intelligence*, pages 199–226.

Nihar B. Shah. 2022. Challenges, experiments, and computational solutions in peer review. *Communications of the ACM*, 65(6):76–87.

Hyungyu Shin, Jingyu Tang, Yoonjoo Lee, Nayoung Kim, Hyunseung Lim, Ji Yong Cho, Hwajung Hong, Moontae Lee, and Juho Kim. 2025. Automatically evaluating the paper reviewing capability of large language models.

Mike Thelwall and Abdullah Yaghi. 2025. Evaluating the predictive capacity of chatgpt for academic peer review outcomes. *Scientometrics*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhe Wei, and Ji-Rong Wen. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6).

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. 2025a. Darwin Gödel machine: Open-ended evolution of self-improving agents. *Preprint*, arXiv:2505.22954.

Jenny Zhang, Bingchen Zhao, Wannan Yang, Jakob Foerster, Jeff Clune, Minqi Jiang, Sam Devlin, and Tatiana Shavrina. 2026. Hyperagents. *Preprint*, arXiv:2603.19461.

Yaohui Zhang, Haijing Zhang, Wenlong Ji, Tianyu Hua, Nick Haber, Hancheng Cao, and Weixin Liang. 2025b. From replication to redesign: Exploring pairwise comparisons for llm-based peer review. *Preprint*, arXiv:2506.11343.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Ruiyang Zhou, Lu Chen, and Kai Yu. 2024. Is llm a reliable reviewer? a comprehensive evaluation of llm on automatic paper reviewing tasks. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pages 9340–9351.

A Additional Implementation Details

This appendix preserves implementation details that are useful for replication and auditing but secondary to the main argument. A creation or tuning run is manually triggered when new venue information, benchmark failures, review-quality observations, or user feedback suggests that an agent version should be repaired or specialized. The factory invokes expert-distilled skills such as sourcing review forms, creating or tuning review agents, sourcing training benchmarks, running benchmarks, inspecting failures, and pushing archived versions. Within a run, the active skill version is fixed; expert changes create a new versioned factory configuration. The serving harness and common utility code remain engineer-maintained; factory skills remain domain-expert-maintained; venue-facing artifacts and training/evaluation sidecars are edited offline.

B Why Borderline Papers Matter

The rating analysis in Figure 2 is not used as a quality leaderboard. Its value is diagnostic. Borderline recommendations are where review assistants must explain uncertainty, separate fatal flaws from fixable presentation issues, and avoid turning a noisy accept/reject proxy into an unjustified conclusion. Low ratings near the threshold therefore provide useful candidates for calibration audits, benchmark refresh, and qualitative failure triage.

C Experimental Scope

The validation task is aligned with a public self-improvement benchmark but is used here as an agent-creation pilot. The same 70/30 train–validation split, base model, candidate scoring budget, evaluation harness, and held-out validation metric are used for SIRA and the HyperAgents-style baseline. Validation examples are not used to generate or repair candidates. The figures show representative traces; Table 2 reports five-run summaries.

D Responsible Use Checklist

A review-agent factory should maintain: (i) separation between private user manuscripts and training data; (ii) explicit versioning of deployed agents; (iii) fixed validation packs plus refreshed diagnostic samples; (iv) human review of low-rated or high-risk failure cases; (v) rollback for promoted versions; and (vi) clear communication that outputs are contestable feedback rather than decisions.