Self-Improving Agent Factories for Paper Review: When Constrained Beats Open-Ended

Lele Cao
CSPaper, a part of Scholar7
Stockholm, Sweden
lele@cspaper.org

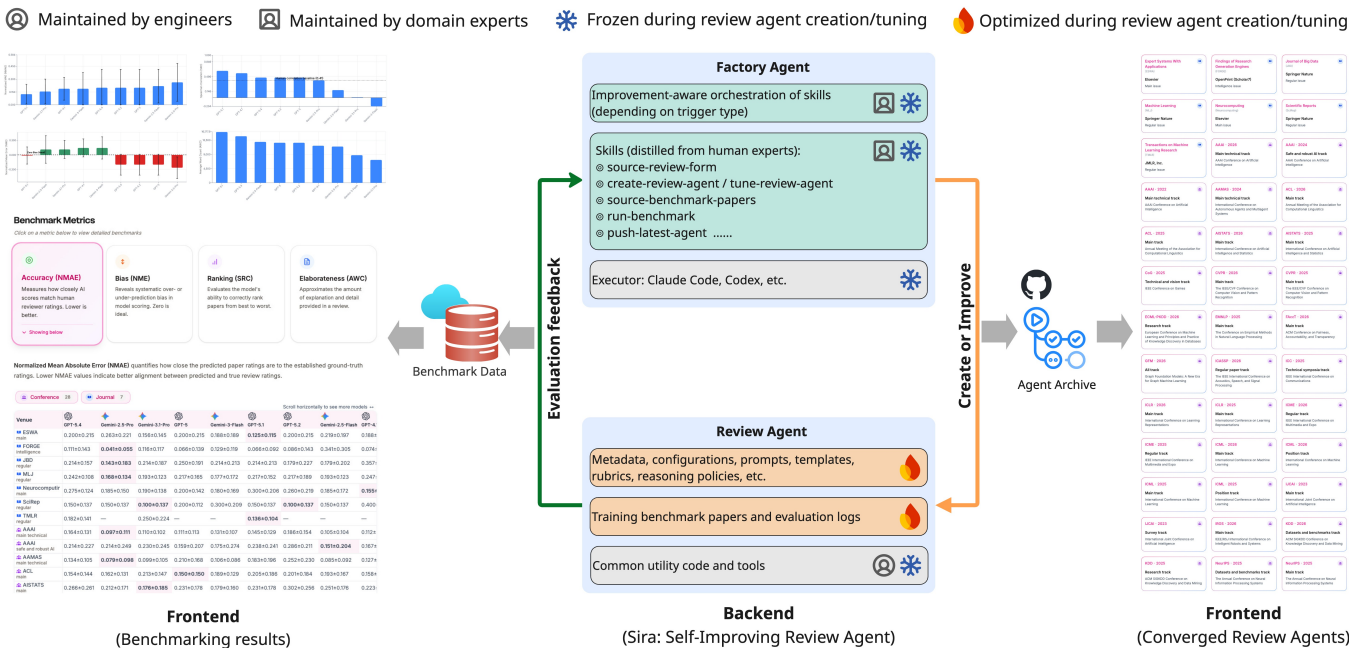


Figure 1: Sira turns self-improvement into an offline, auditable agent-creation loop. Engineers maintain the execution harness and common utilities; domain experts maintain factory skills and review rubrics; the factory evaluates failures and regenerates venue-specific review agents. Only task-specific review-agent artifacts and training benchmark packs are edited; validation packs stay fixed. The deployed review agent is frozen during online service.

Abstract

The default mental model for self-improving agents is increasingly open-ended: let an agent edit itself, edit the procedure that edits itself, and accumulate improvements through exploration. This is scientifically valuable, but often mismatched to production settings where users need predictable, reproducible behavior under privacy, audit, rollback, cost, and service-level constraints. We present Sira, a human-governed self-improving agent-building factory deployed for controlled creation of venue-specific paper-review agents. Sira freezes the factory skills, execution harness, and common utilities during a tuning run, while editing only task-specific artifacts: venue metadata, rubrics, tool use, reasoning policies, prompts, templates, calibration rules, training samples, and evaluation feedback. The created agents are not self-modifying at runtime; they are regenerated, repaired, benchmarked, versioned, and redeployed offline. In a bounded ICLR-style validation-trace study, Sira reached 0.941 ± 0.018 decision-label accuracy within 2-5 scored candidate steps, while a HyperAgents-style mutable-archive adaptation reached 0.865 ± 0.049 within 6-15. This comparison is illustrative rather than causal: Sira intentionally combines a constrained mutability boundary with expert priors such as venue rubrics, calibration templates, and failure-analysis skills, and the metric measures agreement with historical accept/reject labels rather than review quality or untouched test-set generalization. The evidence is limited but operationally meaningful: in this deployed setting, constrained factory regeneration made the improvement process easier to audit, replay, and roll back. We argue that vertical agents with stable rubrics, auditable failures, and privacy-sensitive inputs may often be better built as products of self-improving factories than as free-running self-modifying workers.

Keywords

Agentic AI, self-improving agents, peer review, scientific foundation models, industrial AI systems, evaluation, trustworthiness

1 Industrial Agents Need Factories

AI-assisted peer review is no longer speculative. LLMs are already visible in scientific writing and conference reviewing, and studies

report both useful feedback and serious failure modes. They can provide overlapping but imperfect manuscript feedback [30, 32], yet their correctness judgments, criticism depth, and review substantiation remain unreliable [12, 17, 52, 70]. Decision-side scoring is especially fragile: acceptance prediction, pairwise comparison, and interpretable acceptance models capture useful signals but remain proxy tasks rather than peer review itself [2, 56, 68]. Recent work also documents AI-modified reviews and generated-review datasets, while policy discussions warn against uncritical automation [11, 29, 41, 71]. Meanwhile, CS peer review faces a scale mismatch: AI accelerates paper production, but human review bandwidth remains scarce. CSPaper Review (CSPR)[1] was built for this gap: it accepts arXiv IDs or PDFs and returns conference-aware feedback within roughly one minute, including desk-rejection assessment, expected outcome, and rubric-aligned critical ratings [6]. The question we address is not whether LLM review agents can be useful. It is how such agents should improve in production.

The recent self-improvement literature pushes toward increasingly general self-referential systems. Research on Gödel machines and Darwin-Gödel Machines frames improvement as recursive modification plus validation [49, 66]; HyperAgents extends this idea by letting the meta-improvement mechanism change with the task agent [67]. Adjacent agent work supplies building blocks: reflection and iterative refinement [34, 53], tool use and decomposition [24, 42, 45, 48, 64], retrieval and augmented generation [1, 26, 37], and multi-agent or open-ended task environments [19, 21, 43, 59, 62, 63]. We view this line as important. Our claim is narrower: *many production settings should place self-improvement in a controlled factory that repeatedly rebuilds fixed task agents, rather than in a runtime agent that edits itself.* Runtime self-improvement optimizes for autonomy; industrial deployments must also manage time-to-value, cost, reproducibility, privacy, rollback, and accountable ownership. Peer review makes the tension visible because both over-automation and under-verification are costly: review systems can overfit to human scores and proxy metrics [5, 16, 18, 36, 40, 44], reinforce weak scientific incentives [4, 20, 55], and produce plausible but ungrounded criticism under information and factuality limits [10, 28, 31, 46, 51, 54, 57, 58, 65].

We make **three contributions**. First, we formulate the asymmetric mutability boundary for production agents: the factory may regenerate new task-agent artifacts offline, but the deployed agent does not self-modify while serving users. Second, we report a deployment-grounded case study of this pattern in Sira/CSPR, including how user feedback is converted into offline failure triage rather than online learning. Third, we provide a bounded validation-trace comparison with a mutable-archive baseline to illustrate search behavior under a small scored-candidate budget. The comparison is not an ablation proving that constraint alone beats openness; it evaluates the deployed product pattern of constrained regeneration plus expert priors.

2 Sira: A Review-Agent Factory

Sira is the agent-creation backend used to produce venue-specific CSPR review agents. Its central design choice is an asymmetric mutability boundary: the factory improves agents, but deployed agents do not improve themselves while serving user manuscripts. A review agent is treated as a versioned product artifact rather than a continuously learning worker. Concretely, each venue build contains a deployed agent

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

where m_v denotes 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 shared utility code inherited from the CSPR serving harness. A separate sidecar b_v contains training benchmark papers, expected labels, and evaluation records; it is not deployed with the agent. Sira modifies m_v, r_v, p_v, t_v, c_v, and b_v during creation/tuning, while u and the factory controller are frozen during a run and maintained separately by engineers and domain experts. A factory improvement step can be written as

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

where F_ϕ is the frozen factory agent, E is the evaluation harness, S_v is the expert-maintained skill library, and H_v captures explicit human guidance. The allowed actions are intentionally mundane: source official venue information, create or tune a review agent, curate training benchmark papers, run evaluation, inspect failures, and push a named version to the archive. The point is not to prevent improvement; it is to make the locus of improvement inspectable. Sira edits the parts of the system that should vary with venue and evidence, while leaving the execution harness and online service behavior reproducible.

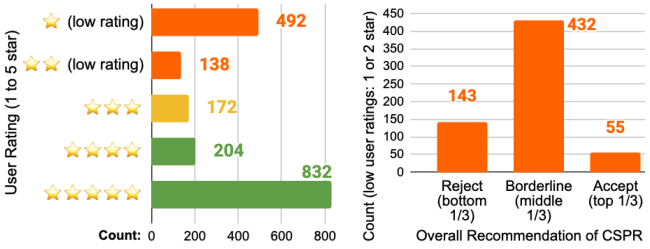
Concretely, each tuning run follows the same sequence. (1) A human trigger, such as new venue instructions, a benchmark failure, or a low-rating audit ticket, opens a versioned run. (2) The frozen factory F_ϕ materializes a candidate by editing only m_v, r_v, p_v, t_v, c_v and the training sidecar b_v; it cannot edit the executor, common utilities u, evaluation scripts, validation set, or its own controller. (3) The candidate is scored on training benchmark packs and inspected through rubric-level failure summaries; optional human guidance H_v is recorded as part of the run. (4) Only after the candidate is materialized, the fixed evaluator scores it on the validation split for reporting and archive comparison. Validation examples and labels are not copied into prompts, sidecars, or human guidance. (5) A promoted candidate is pushed as a named archive version, and the online service loads that frozen bundle until a later offline run replaces it. Thus, "self-improving" here means factory-mediated regeneration under fixed skills and logged human governance, not autonomous runtime learning from private submissions.

In **production**, creation or tuning is manually triggered by engineers or domain experts when new venue information, benchmark failures, review-quality observations, or user feedback suggests that an agent version should be repaired or specialized. Sira then invokes expert-distilled skills (e.g., sourcing review forms, creating or tuning agents, sourcing training benchmarks, running benchmarks, inspecting failures, and pushing archived versions) to regenerate the task-specific bundle. The serving harness and common utility code remain engineer-maintained; factory skills remain domain-expert-maintained; only venue-facing artifacts and training/evaluation sidecars are edited.

[1] CSPR (https://cspaper.org) processed over 150,000 review requests from 130 countries at the time of writing, according to aggregate production telemetry.

Table 1: Design contrast. Sira sacrifices some open-endedness to satisfy production constraints that matter for unpublished scientific manuscripts: reproducibility, privacy, rollback, and cost control.

Dimension	Open editable-agent baseline	Sira factory regeneration	Practical implication
Editable object	Task behavior and the procedure that generates future task behavior may both change.	Factory skills and common utilities are frozen during a run; only venue-specific review-agent artifacts and benchmark packs evolve.	Stronger auditability and cleaner ownership between engineers, domain experts, and deployed agents.
Improvement site	Improvement can be coupled to the live or fully autonomous agent loop.	Improvement is offline: regenerate, evaluate, version, and redeploy.	Live research manuscripts are processed by fixed agents, improving reproducibility and rollback.
Search bias	Broad exploration can discover surprising mechanisms but may spend budget rediscovering domain structure.	Expert-constrained search begins with official rubrics, known failure modes, calibrated examples & benchmark diagnostics.	Can reduce search effort when the target is a narrow vertical product rather than open-ended discovery.
Evidence used for promotion	Any feedback the agent can exploit, often entangled with evolving meta-procedures.	Fixed validation packs, refreshed benchmark samples, rubric-level diagnostics, and human failure triage.	Promotion decisions are easier to replay and contest.
Risk profile	More expressive but harder to bound when the self-modification mechanism changes.	Less general but designed for privacy, governance, and predictable service behavior.	Appropriate for high-volume pre-review, not a replacement for human reviews.



(a) Rating distribution **(b) Low-rating breakdown**

Figure 2: Post-review quality ratings for CSPR outputs collected after delivery (n = 1,838 ratings; 630 one- or two-star ratings). Low-rated reviews are not used as online training data; when policy permits, they are audited as de-identified operational signals for revision utility, failure triage, and benchmark refresh. In this snapshot, low ratings concentrate most heavily around borderline recommendations, suggesting that near-threshold papers are useful for diagnosing calibration and feedback-quality failures.

User **feedback** on delivered reviews is an especially useful operational signal because it reveals failures that benchmark accuracy alone may miss. As shown in Figure 2, CSPR collects post-review quality ratings and audits low-rated cases by their overall recommendation scores [7]. These ratings are not online training data. When policy permits, low-rated reviews become de-identified tickets, aggregate diagnostics, or benchmark-refresh candidates that engineers and domain experts inspect before any offline tuning run. This conservative boundary matches peer review's trust requirements: private manuscripts may reveal where the system fails, but they should not silently alter the system that reviews the next research manuscript. This telemetry motivates offline triage and benchmark refresh; it is not used below as evidence that Sira produces higher-quality critiques than the baseline.

3 Comparative Study: Sira vs. HyperAgents

We evaluate Sira in the same *paper-review domain* studied by HyperAgents [67], but with a different split protocol for an agent-creation pilot rather than a leaderboard reproduction. Each sample contains the full text of an AI research paper and a binary accept/reject label derived from real conference decisions. This label is a decision-side proxy: it measures agreement with historical outcomes, not the factual correctness or usefulness of generated critiques. We re-split the available paper-review dataset into a fixed 70/30 train–validation partition, keeping the split and class balance unchanged across all runs. The training split is used inside the creation loop: Sira creates, repairs, and evolves candidate ICLR-style CSPR review agents using training feedback, benchmark failures, and human-inspected observations. The validation split is kept outside candidate construction and used to score and report the generated candidates and the search trajectory in Figure 3. The operational question is how this constrained, auditable factory behaves under a small scored-candidate budget.

The 70/30 split is used as a train–validation split, not as a train–validation–test protocol. Training papers and labels may be used inside the creation loop through b_v. Validation papers are held out from candidate construction: their text and labels are not included in b_v, prompts, benchmark packs, or human guidance. However, validation accuracy is the scoring signal used to plot and compare candidate trajectories, so "best validation accuracy" is a selected-on-validation statistic rather than an untouched test estimate. We also do not report a near-duplicate/resubmission audit, temporal split, or base-model pretraining cutoff relative to the paper decisions. The numbers should therefore be read as within-protocol historical decision-label alignment, not as fresh-conference generalization.

The baseline is a HyperAgents-style adaptation in which both task-agent behavior and the mutation policy that proposes future candidates are represented in an editable archive. It is not an official reproduction of HyperAgents, and we make no claim about the original system. The baseline and Sira share the base model, split,

Table 2: Five-run validation summary on the re-split paper-review dataset. Both settings are repeated five times on the same 70/30 train–validation partition.

Performance metric (↑/↓ better)	Sira	HyperAgents-style
Best validation decision-label accuracy (↑)	**0.941 ± 0.018**	0.865 ± 0.049
Final archive mean decision-label accuracy (↑)	**0.868 ± 0.023**	0.590 ± 0.072
Scored candidate steps to best validation score (min–max ↓)	**2–5**	6–15
Mutable meta agent during run	No	Yes

Notes. Values with ± denote mean ± run-to-run standard deviation over five repeated runs. Accuracy is binary agreement with historical accept/reject labels; it is not a measure of factuality, novelty assessment, review helpfulness, fairness, or calibration. The validation split is held out from candidate construction but used for trajectory scoring, so the best-validation row is selected on validation and should not be interpreted as untouched test performance. The final-archive mean is an archive-stability diagnostic, not a final-agent capability metric, because exploratory methods intentionally retain weak intermediate candidates.

evaluator, scoring script, and candidate-scoring budget. They do not isolate a single causal variable: Sira's deployed pattern intentionally bundles constrained mutability with expert-maintained priors (S_v), while the mutable-archive baseline is allowed broader structural changes but does not use the same hand-shaped factory skills. The comparison therefore asks a product question: how the Sira pattern behaves under a small budget relative to an open mutable-archive adaptation, not whether constraints alone are universally better than open-ended self-improvement.

The five-run results support a bounded operational reading. Under this protocol, Sira found a higher-scoring validation candidate and reached it with fewer scored candidate steps. The final-archive mean should be read only as a stability diagnostic: it indicates that Sira's constrained archive contained fewer low-scoring exploratory candidates, not that every candidate in the archive was a better deployable agent. The likely explanation is search compression from human priors (official rubrics, known review failure modes, score-calibrated templates, and benchmark-aware failure analysis) rather than greater generality. Open-ended systems can in principle discover such structure; the product question is whether to pay that exploration cost when the structure is already available from domain experts.

This distinction can be expressed as an engineering objective rather than a philosophical preference:

$$\arg\max_{a} \; U_v(a) - \lambda C(a) - \rho R(a) - \gamma T(a), \tag{3}$$

where $U_v(a)$ is venue-specific utility, $C(a)$ is compute, engineering, and human-curation cost, $R(a)$ is operational risk such as privacy, leakage, auditability, and rollback risk, and $T(a)$ is time-to-deploy. In this paper, U_v is only partially proxied by validation decision-label accuracy, and scored candidate steps are only a rough search-effort proxy rather than a full measurement of C or T. We do not report token cost, wall-clock tuning time, serving latency, or expert-hour cost for maintaining S_v; Eq. (3) is therefore an engineering objective that motivates the design, not a fully measured result. A more general self-improving agent may eventually improve U_v, but a factory can be preferable when it reduces the operational terms enough for a production setting.

4 Scientific and Industrial Implications

The comparison above highlights several design observations for scientific and industrial self-improving agents.

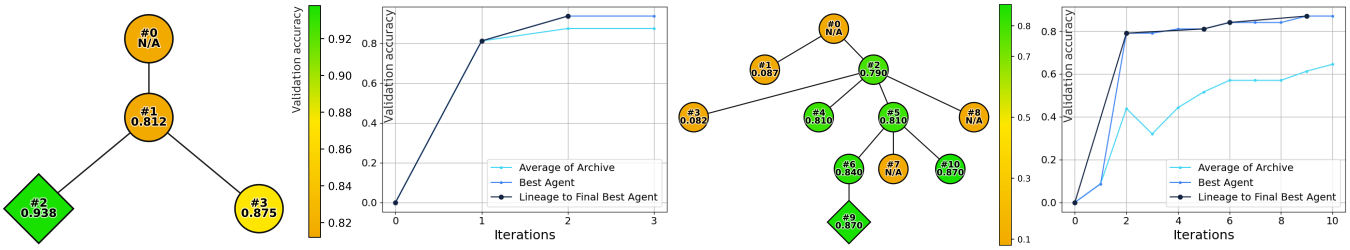
Put self-improvement at the artifact boundary. A review agent serving an unpublished manuscript should be stable enough to debug, versioned enough to cite, and frozen enough to reproduce. Improvement belongs in the factory that creates the next version. This resembles continuous integration for software, except the tested unit is an agent bundle containing rubrics, retrieval policies, prompts, templates, calibration rules, and evaluation packs.

Treat evaluation as product infrastructure. Sira does not promote an agent because its reviews sound more polished. Promotion is tied to fixed validation packs, refreshed training benchmark packs, rubric-level diagnostics, and qualitative failure audits. This connects to work on evidence grounding and retrieval [13, 57], truthful and factual generation [31, 35, 38], model and dataset reporting [3, 14, 39], generated-text detection, reasoning, and uncertainty-aware behavior [15, 22, 61], and broad model and agent evaluation [8, 28, 33, 60, 69]. For scientific agents, evaluation is not an appendix; it is part of the deployed object.

Use human expertise as search compression, not merely supervision. The factory is dominated by human-maintained skills because venue expertise is itself a high-value prior. A domain expert who encodes review rubrics, common failure modes, and calibration examples can remove many unpromising branches before exploration starts. This does not make the system less scientific. It makes the optimization target more honest: the goal is not maximal autonomy, but reliable task performance under cost, privacy, and governance constraints.

Produce contestable artifacts, not final judgments. CSPR and Sira are designed for pre-submission feedback and computational research assessment, not for replacing program committees. The output should be an evidence-linked, rubric-grounded critique that authors and reviewers can challenge. This matters because peer review is a social decision process: assignment and bidding shape who evaluates what [47], outcomes contain arbitrariness and inconsistency [9, 25, 50], and datasets or models capture only parts of that process [23, 27]. A system that merely imitates historical decisions can preserve the appearance of review while missing the epistemic work of verification.

Scope of transfer. Our evidence is strongest for domains that resemble the paper-review setting studied here: the task has public or stable instructions, expert-maintained rubrics, versionable artifacts, repeatable evaluation packs, and high cost for uncontrolled online adaptation. We do not show that the same advantage holds



(a) Sira search tree (b) Sira progress (c) HyperAgents-style tree (d) HyperAgents-style progress

Figure 3: Representative self-improvement traces on the paper-review validation task. Nodes show archive candidates; colors encode validation accuracy when available; N/A marks unscored nodes; diamonds mark the final best candidate. This run illustrates the behavior summarized over five runs in Table 2: Sira reaches a high-scoring validation candidate with fewer scored candidates, consistent with the interpretation that venue rubrics, expert skills, and benchmark-aware failure analysis pre-shape the search space.

when these conditions are absent, for example when objectives change quickly, feedback cannot be audited, or useful priors must be discovered from scratch.

5 Limitations and Responsible Use

This paper should be read as a position paper grounded in a deployed system, with a small validation-trace study used to illustrate the design rather than to settle a benchmark. The trace supports a practical claim: in one review-agent creation setting, a governed factory with expert priors found a high-scoring validation candidate in 2–5 scored candidate steps while preserving an auditable artifact boundary. It does not isolate which ingredient caused the gain. The effect of constrained mutability is entangled with venue rubrics, calibration templates, benchmark curation, and human-inspected failure analysis. Likewise, binary agreement with historical accept/reject labels is only a decision-side signal. It is useful for debugging calibration, but it is not review quality: it does not measure factuality, depth of criticism, novelty assessment, author usefulness, or fairness across topics, institutions, and regions.

The next empirical step is to make the factory pattern more measurable: evaluate across more venues, add untouched final test sets, run matched-prior ablations, log per-iteration artifact diffs, and report token cost, latency, and expert time. Until then, Sira should be interpreted as an operational pattern for privacy-sensitive vertical agents, not as evidence that constrained self-improvement universally dominates open-ended self-modification.

Responsible deployment also requires a strict product boundary. Review agents should not be used as authoritative acceptance predictors, should not silently learn from private submissions, and should not be presented as substitutes for human reviewers. Their role is earlier and narrower: help authors identify missing evidence, weak claims, unclear contributions, benchmark gaps, and rubric mismatches before submission. In that role, the factory pattern provides a pragmatic compromise between adaptation and control.

6 Conclusion

Open-ended self-improving agents ask how far agents can go if they rewrite themselves. Sira asks a complementary industrial question: how much value can be delivered when agents are built and tuned by a disciplined factory? In the CSPR deployment, this factory pattern provided a practical way to regenerate venue-specific review agents while keeping a clean audit boundary between factory, benchmark, and online service. The broader lesson is not that generality is bad. It is that generality should be placed where it pays for itself. For vertical agents with stable rubrics, auditable failures, and privacy-sensitive inputs, the frontier may be less about free-running autonomy and more about auditable production lines that know exactly what not to let change.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. arXiv:2310.11511 [cs.CL]

[2] Peng Bao, Weihui Hong, and Xuanya Li. 2021. Predicting Paper Acceptance via Interpretable Decision Sets. In *Companion Proceedings of the Web Conference 2021*. 461–467.

[3] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 610–623.

[4] Nicholas Bloom, Charles I. Jones, John Van Reenen, and Michael Webb. 2020. Are Ideas Getting Harder to Find? *American Economic Review* 110, 4 (2020), 1104–1144.

[5] Donald T. Campbell. 1979. Assessing the Impact of Planned Social Change. *Evaluation and Program Planning* 2, 1 (1979), 67–90.

[6] 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*. Association for Computational Linguistics, Hanoi, Vietnam, 3–7. https://aclanthology.org/2025.inlg-demos.2/

[7] 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* (2026). https://cspaper.org/openprint/20260212.0002v1

[8] 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, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374 [cs.LG]

[9] Corinna Cortes and Neil D. Lawrence. 2021. Inconsistency in Conference Peer Review: Revisiting the 2014 NeurIPS Experiment. In *NeurIPS 2021 Workshop on*

Meta-Learning.
[10] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. Wiley.
[11] Luca Demetrio, Giovanni Apruzzese, Kathrin Grosse, Pavel Laskov, Emil Lupu, Vera Rimmer, and Philine Widmer. 2025. Gen-Review: A Large-scale Dataset of AI-Generated (and Human-written) Peer Reviews. arXiv:2510.21192 [cs.LG] doi:10.48550/arXiv.2510.21192
[12] 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. arXiv:2409.05367 [cs.CL] doi:10.48550/arXiv.2409.05367 Accepted at ACL 2025.
[13] 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* (2023).
[14] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for Datasets. *Commun. ACM* 64, 12 (2021), 86–92.
[15] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: Statistical Detection and Visualization of Generated Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 111–116.
[16] Charles A. E. Goodhart. 1984. Problems of Monetary Management: The UK Experience. In *Monetary Theory and Practice*. Macmillan, London, 91–121.
[17] 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*. 10198–10216.
[18] Bengt Holmstrom and Paul Milgrom. 1991. Multitask Principal-Agent Analyses: Incentive Contracts, Asset Ownership, and Job Design. *Journal of Law, Economics, and Organization* 7 (1991), 24–52.
[19] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Ceyao Wang, Zili Zhang, Steven K. S. Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, and Chenglin Wu. 2023. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. arXiv:2308.00352 [cs.AI]
[20] John P. A. Ioannidis. 2005. Why Most Published Research Findings Are False. *PLoS Medicine* 2, 8 (2005), e124.
[21] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can Language Models Resolve Real-World GitHub Issues?. In *International Conference on Learning Representations*.
[22] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language Models (Mostly) Know What They Know. arXiv:2207.05221 [cs.CL]
[23] 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*. 1647–1661.
[24] Eyal Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hila Bata, Yoav Levine, Kevin Leyton-Brown, Dor Muhlgay, Neer Rozen, Roy Schwartz, and Gal Shachaf. 2022. MRKL Systems: A Modular, Neuro-Symbolic Architecture That Combines Large Language Models, External Knowledge Sources and Discrete Reasoning. arXiv:2205.00445 [cs.CL]
[25] John Langford and Mark Guzdial. 2015. The Arbitrariness of Reviews, and Advice for School Administrators. Communications of the ACM Blog.
[26] 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*, Vol. 33. 9459–9474.
[27] 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*. 121–126.
[28] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic Evaluation of Language Models. arXiv:2211.09110 [cs.CL]
[29] Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, Daniel A. McFarland, and James Zou. 2024. Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews. *Proceedings of the 41st International Conference on Machine Learning* (2024).
[30] Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, et al. 2024. Can Large Language Models Provide Useful Feedback on Research Papers? A Large-Scale Empirical Analysis. *NEJM AI* 1, 8 (2024).
[31] 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*. 3214–3252.
[32] Ryan Liu and Nihar B. Shah. 2023. ReviewerGPT? An Exploratory Study on Using Large Language Models for Paper Reviewing. arXiv:2306.00622 [cs.CL]
[33] 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*. 2511–2522.
[34] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. arXiv:2303.17651 [cs.CL]
[35] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. arXiv:2303.08896 [cs.CL]
[36] David Manheim and Scott Garrabrant. 2019. Categorizing Variants of Goodhart's Law. arXiv:1803.04585 [cs.AI]
[37] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented Language Models: a Survey. *Transactions on Machine Learning Research* (2023).
[38] 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*. 12076–12100.
[39] 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*. 220–229.
[40] Jerry Z. Muller. 2018. *The Tyranny of Metrics*. Princeton University Press.
[41] Miryam Naddaf. 2025. AI is Transforming Peer Review – and Many Scientists Are Worried. *Nature* 639, 8056 (2025), 852–854.
[42] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332 [cs.CL]
[43] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*.
[44] Michael Power. 1997. *The Audit Society: Rituals of Verification*. Oxford University Press.
[45] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and Narrowing the Compositionality Gap in Language Models. arXiv:2210.03350 [cs.CL]
[46] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling Language Models: Methods, Analysis and Insights from Training Gopher. arXiv:2112.11446 [cs.CL]
[47] Mark Roos, Jorg Rothe, and Bjorn Scheuermann. 2011. Reviewer Assignment Problem: A Systematic Review of the Literature. *Journal of Artificial Intelligence Research* 42 (2011), 523–545.
[48] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. arXiv:2302.04761 [cs.CL]
[49] Jurgen Schmidhuber. 2007. Godel Machines: Fully Self-Referential Optimal Universal Self-Improvers. *Artificial General Intelligence* (2007), 199–226.
[50] Nihar B. Shah. 2022. Challenges, Experiments, and Computational Solutions in Peer Review. *Commun. ACM* 65, 6 (2022), 76–87.
[51] Claude E. Shannon. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* 27, 3 (1948), 379–423.
[52] 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. arXiv–2502 pages.
[53] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366 [cs.AI]
[54] Christopher A. Sims. 2003. Implications of Rational Inattention. *Journal of Monetary Economics* 50, 3 (2003), 665–690.
[55] Paul E. Smaldino and Richard McElreath. 2016. The Natural Selection of Bad Science. *Royal Society Open Science* 3, 9 (2016).
[56] Mike Thelwall and Abdullah Yaghi. 2025. Evaluating the Predictive Capacity of ChatGPT for Academic Peer Review Outcomes. *Scientometrics* (2025).
[57] 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). 809–819.

[58] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. FEVER 2.0: An Open Challenge for Fact Extraction and Verification. In *Proceedings of the Second Workshop on Fact Extraction and VERification*. 1–6.

[59] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research* (2024).

[60] 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 (2024).

[61] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 35. 24824–24837.

[62] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. arXiv:2308.08155 [cs.AI]

[63] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. arXiv:2405.15793 [cs.SE]

[64] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*.

[65] Lei You, Lele Cao, and Iryna Gurevych. 2026. Preventing the Collapse of Peer Review Requires Verification-First AI. arXiv:2601.16909 [cs.AI] doi:10.48550/arXiv.2601.16909

[66] Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. 2025. Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents. arXiv:2505.22954 [cs.AI] doi:10.48550/arXiv.2505.22954

[67] Jenny Zhang, Bingchen Zhao, Wannan Yang, Jakob Foerster, Jeff Clune, Minqi Jiang, Sam Devlin, and Tatiana Shavrina. 2026. Hyperagents. arXiv:2603.19461 [cs.AI] doi:10.48550/arXiv.2603.19461

[68] Yaohui Zhang, Haijing Zhang, Wenlong Ji, Tianyu Hua, Nick Haber, Hancheng Cao, and Weixin Liang. 2025. From Replication to Redesign: Exploring Pairwise Comparisons for LLM-Based Peer Review. arXiv:2506.11343 [cs.CL]

[69] 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. arXiv:2306.05685 [cs.CL]

[70] 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)*. 9340–9351.

[71] James Zou. 2024. ChatGPT is transforming peer review—how can we use it responsibly. *Nature* 635, 8037 (2024), 10–10.