ReproEvalCard: A Reporting Standard for Reproducible Evaluation of LLM Pipelines

Priyaranjan Pattnayak[1], **Apoorv Bhatia**[2]
[1]Oracle America Inc. [2]Oracle Canada.
Correspondence: priyaranjanpattnayak@gmail.com

Abstract

Evaluation of modern large language model (LLM) systems increasingly relies on multi-stage pipelines such as retrieval-augmented generation, tool-using agents, and prompt chains. Reproducing reported evaluation results for these systems often requires evaluation-specific artifacts beyond model weights and datasets, including prompts, judge configurations, retrieval snapshots, and intermediate traces, yet their availability has not been systematically examined.

We introduce **ReproEvalCard**, a lightweight reporting standard that specifies the minimal artifacts required to reproduce and validate evaluations of LLM pipelines. To motivate this standard, we audit 55 pipeline-based LLM papers published between 2022 and 2025 and quantify the availability of reproducibility-critical evaluation artifacts. We find that randomness controls are missing in 75% of papers and intermediate execution traces in 61%, substantially limiting evaluation reproducibility. We further demonstrate ReproEvalCard through a worked example and provide a concise checklist for authors and reviewers, aiming to improve reproducibility and comparability in LLM evaluation.

1 Introduction

Evaluation is central to progress in large language model (LLM) research. Increasingly, state-of-the-art systems are implemented as multi-stage pipelines rather than single-model predictors, including retrieval-augmented generation (RAG), tool-using agents, prompt chains, and planner–executor architectures (Lewis et al., 2020; Yao et al., 2022). In these systems, reported evaluation outcomes depend not only on the underlying model but also on intermediate components and configuration choices.

Reproducing LLM pipeline evaluations therefore requires access to evaluation-specific artifacts such as generation and evaluation prompts, judge model configurations, retrieval corpus snapshots, tool schemas, and intermediate execution traces. While some artifacts may be released via code or supplementary material, their availability and completeness are inconsistent across papers, leaving the reproducibility and independent validation of published evaluations unclear.

Prior work has proposed general reproducibility checklists and artifact evaluation processes, primarily focused on model training, datasets, and code release (Pineau et al., 2021). More recently, standardized formats for evaluation reporting have been proposed to improve transparency (Dhar et al., 2025). However, these approaches do not explicitly target the reproducibility requirements of evaluation execution in multi-stage LLM pipelines, nor do they empirically assess whether the necessary artifacts are reported in current practice.

In this work, we focus on evaluation reproducibility for LLM pipelines. We introduce **ReproEvalCard**, a concise reporting standard that captures the minimal set of artifacts required to reproduce and validate evaluation results for pipeline-based LLM systems. To ground this proposal in evidence, we conduct a systematic audit of **55** recent LLM pipeline papers published between 2022 and 2025. Our analysis reveals substantial reproducibility risks, including **75% of papers missing randomness controls** and frequent absence of execution-critical artifacts. These findings highlight recurring gaps in evaluation reporting.

Positioning. To our knowledge, this is the first systematic audit of reproducibility artifacts in modern multi-stage LLM evaluation pipelines. We focus on the *evaluation execution layer* (generation, retrieval, tool use, judging, and aggregation), where missing specifications directly prevent reconstruction of reported results and remain largely unaddressed by prior documentation standards.

Our contributions are as follows:

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 238–249

- We introduce **ReproEvalCard**, a reproducibility-focused reporting standard for evaluation of multi-stage LLM pipelines.
- We present a **systematic audit of 55 LLM pipeline papers (2022–2025)**, quantifying the availability of execution-critical evaluation artifacts.
- We provide **empirical evidence of reproducibility gaps** (e.g., 75% missing randomness controls), and include cross-pipeline reconstruction case studies.
- We demonstrate the application of ReproEvalCard through a worked example and provide a minimal checklist for authors and reviewers.

By explicitly targeting evaluation execution rather than model or dataset properties, our work complements existing reproducibility efforts and provides a practical tool for improving evaluation reliability in modern LLM systems.

2 Related Work

Reproducibility in Machine Learning. Reproducibility concerns have prompted artifact evaluations and reproducibility checklists across machine learning venues (Pineau et al., 2021). These efforts focus on training procedures, datasets, and code availability but are largely agnostic to evaluation execution and overlook evaluation-specific artifacts in multi-stage LLM pipelines.

Evaluation Reporting and Documentation. Several documentation frameworks, including Model Cards and Data Cards, aim to standardize disclosure of model and dataset properties (Mitchell et al., 2019; Gebru et al., 2021). Concurrent work by Dhar et al. (2025) proposes *EvalCards*, a standardized format for evaluation reporting intended to improve transparency and accessibility. EvalCards focus on broad evaluation disclosure across tasks and settings, whereas our work targets the reproducibility of the *evaluation execution layer* in multi-stage LLM pipelines, emphasizing the concrete artifacts required to re-run and validate reported results.

Evaluation of LLM Pipelines. A growing body of work examines evaluation methods for retrieval-augmented generation, agentic systems, and multi-step prompting (Lewis et al., 2020; Yao et al., 2022). While these studies often propose new benchmarks or metrics, they typically assume the availability of evaluation artifacts without scrutinizing how they are reported. In contrast, our work audits existing reporting practices and proposes a minimal standard to improve evaluation reproducibility.

3 Audit Method

To assess evaluation reporting practices for LLM pipelines, we conducted a systematic audit of recent papers evaluating multi-stage LLM systems. Rather than assessing model performance, we examined the availability of evaluation artifacts critical for reproducing or validating reported results.

Paper Selection. We analyzed 55 papers published between 2022 and 2025 from ACL, EMNLP, NAACL, EACL, ICLR, NeurIPS, ICML, and arXiv. Papers were included if they evaluated pipeline-based LLM systems (e.g., retrieval-augmented generation, tool-using agents, or prompt chains) and reported quantitative results. We excluded work focused solely on pretraining, theory, or prompt-only ablations. The full paper list appears in Appendix B.

Artifact Coding. Each paper was independently coded for the presence of eight evaluation artifacts needed to reproduce evaluation execution: generation and judge prompts, judge model specifications, retrieval corpora or index snapshots, tool schemas or APIs, intermediate traces, evaluation scripts or protocols, and randomness controls. Artifacts were coded as *Present*, *Partial*, or *Missing* using explicit criteria (Appendix A), with ambiguous cases conservatively labeled *Missing*.

Annotation reliability. Artifact labels were independently assigned by two annotators using the rubric in Appendix A. We observe substantial agreement (macro-averaged Cohen's $\kappa = 0.78$), with disagreements primarily in borderline *Partial* vs. *Missing* cases. Detailed per-category agreement is provided in Appendix E.

Analysis. We report aggregate availability rates and system-level breakdowns for each artifact. The analysis is descriptive, aiming to characterize current reporting practices rather than rank papers or make normative claims.

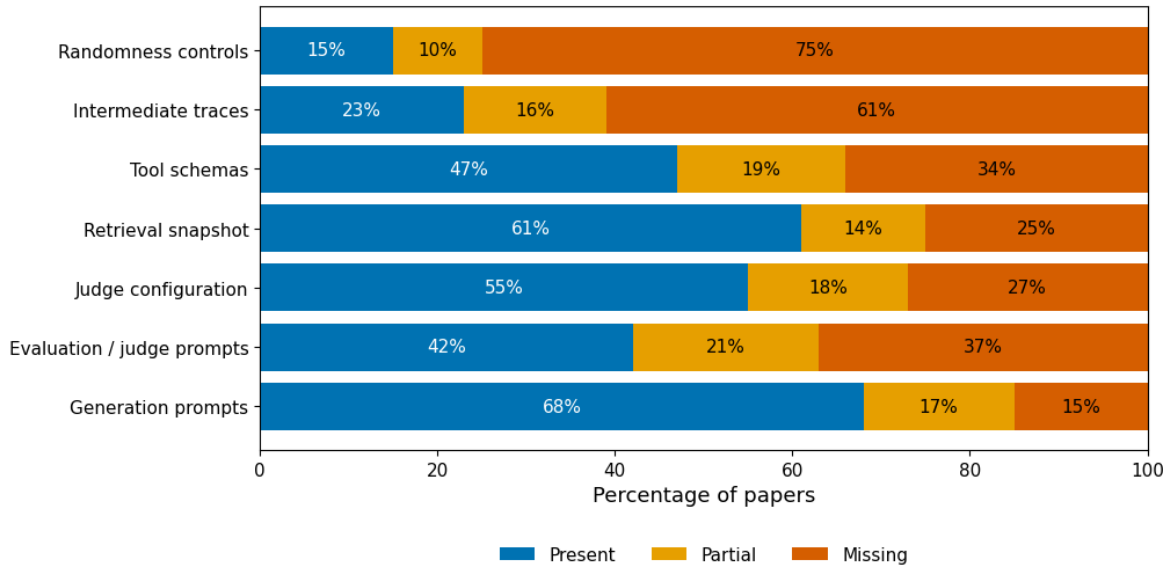

Figure 1: Availability of reproducibility-critical evaluation artifacts across 55 recent LLM pipeline papers.

4 Results

This section presents empirical findings from our audit of evaluation artifact reporting in pipeline-based LLM systems. We analyze overall artifact availability, variation across pipeline types, and dominant sources of irreproducibility.

4.1 Overall Evaluation Artifact Availability

We first examine the availability of reproducibility-critical evaluation artifacts across all audited papers. Fig 1 summarizes the proportion of papers in which each artifact is present, partially specified, or missing.

Generation prompts and retrieval corpus descriptions are reported in a majority of papers. In contrast, several artifacts required to reproduce evaluation execution are frequently absent. In particular, intermediate execution traces and randomness controls are missing in most evaluations, limiting the ability to independently validate reported results even when code or datasets are released.

4.2 Variation by Pipeline Type

Artifact availability is not uniform across LLM pipeline types. Table 1 presents a breakdown of selected evaluation artifacts by pipeline category.

Agent-based systems more frequently report partial tool configurations and intermediate traces, while retrieval-augmented generation systems more often specify retrieval corpora but omit randomness controls. Prompt-chain systems consistently underreport evaluation prompts and judge configurations. These differences indicate that evaluation reproducibility challenges depend on pipeline structure rather than arising uniformly across systems.

Artifact	RAG	Agents	Prompt Chains
Generation prompts	72%	64%	69%
Evaluation prompts	48%	31%	44%
Judge configuration	59%	42%	53%
Intermediate traces	18%	29%	21%
Randomness controls	12%	17%	14%

Table 1: Availability of selected evaluation artifacts by pipeline type (percent present). Shading is proportional to percentage values (0–100).

4.3 Dominant Sources of Evaluation Opacity

Beyond per-artifact availability, we identify the dominant source of evaluation opacity for each paper, defined as the single most consequential missing or underspecified element that prevents faithful reproduction of evaluation execution. Table 2 summarizes these dominant obstacles.

Missing prompt templates for generation is the most common dominant obstacle, followed by unspecified evaluation setups such as unclear judge prompts, criteria, or judge model settings. Missing intermediate steps or outputs also frequently prevents validation, especially for tool-using and multi-step pipelines.

4.4 Reproducibility Risk Index

To summarize artifact availability at the paper level, we define a simple *Reproducibility Risk Index (RRI)* as the number of reproducibility-critical artifacts

Dominant Source of Opacity	% of Papers
Missing prompt templates (generation prompts)	40
Unspecified evaluation setup (judge prompt/criteria/config)	25
Missing intermediate steps or outputs	15
Unspecified tool or environment details	10
Uncontrolled nondeterminism (single run, no seed)	10

Table 2: Dominant sources of evaluation opacity across audited papers. Percentages reflect the primary obstacle per paper.

reported for evaluation execution:

$$\mathrm{RRI}(p) = \sum_{a \in \mathcal{A}} \mathbb{1}\left[\text{artifact } a \text{ is present in paper } p\right], \tag{1}$$

where $\mathcal{A}$ is the set of audited evaluation artifacts (Section 3). Higher RRI indicates lower evaluation reproducibility risk. RRI uses uniform weighting, since any single missing execution-critical artifact can block reproducibility.

Sensitivity to Partial. To assess robustness, we evaluate RRI under alternative treatment of *Partial* artifacts (strict: Partial = 0; soft: Partial = 0.5). We observe high rank correlation (Spearman $\rho = 0.91$) with no change in pipeline ordering, indicating stable conclusions.

Table 3 reports the distribution of RRI across papers and average RRI by pipeline type. Most audited papers fall into the medium-risk range, with agent and prompt-chain systems exhibiting lower average RRI than RAG systems.

4.5 Summary of Empirical Findings

Taken together, Fig 1, Tables 1, 2, and 3 show that evaluation reporting often omits execution-critical artifacts, with systematic variation by pipeline type and a small set of recurring dominant obstacles. These results motivate a concise, evaluation-focused reporting standard.

Beyond reproducibility, the missing artifacts identified in Section 4 also undermine peer-review reliability. Lacking access to evaluation prompts, judge configurations, randomness controls, or intermediate execution traces, reviewers must rely on high-level descriptions to assess reported results. By making these evaluation assumptions explicit and centralized, ReproEvalCard supports both post-hoc reproduction and more informed, consistent peer review.

RRI Range	Overall (%)	RAG (%)	Agents (%)	Chains (%)
0–2 (High risk)	35	20	45	50
3–5 (Medium)	55	65	50	45
6–8 (Low risk)	10	15	5	5
Avg. RRI	**3.2**	**3.7**	**2.8**	**2.5**

Table 3: Reproducibility Risk Index (RRI) distribution overall and by pipeline type. Higher RRI indicates more evaluation artifacts reported.

Artifact	Present	Partial	Missing	Block %
Base model spec.	0	18	2	10
Decoding params.	0	14	6	60
Generation prompt	0	12	8	60
Evaluation protocol	4	16	0	20
Retrieval snapshot	4	10	6	40
Tool/API spec.	0	10	10	30
Randomness controls	4	6	10	**80**

Table 4: Artifact-level reconstruction breakdown for $N = 20$ papers. Color intensity reflects relative magnitude within each column (green = present, yellow = partial, red = missing/blocking).

Across all analyses, the most severe reproducibility risks stem not from model or dataset opacity but from evaluation execution details, particularly nondeterminism and missing intermediate traces, suggesting that existing reproducibility efforts focused on code and data release overlook the dominant failure modes in modern LLM pipeline evaluations.

4.6 Reconstruction Feasibility from Published Artifacts

To assess the impact of missing evaluation artifacts, we conduct a reconstruction feasibility study on a stratified subset of $N = 20$ audited papers spanning RAG, agent, and prompt-chain pipelines. For each paper, we attempt to reconstruct the evaluation setup using only information available in the published paper and appendix, without relying on external code or supplementary repositories.

Reconstruction is considered successful only if all execution-critical components (model configuration, prompts, decoding parameters, retrieval state, and evaluation protocol) are fully specified. We record the primary missing artifact that prevents reconstruction under this strict criterion.

We find that reconstruction succeeds in only 2 out of 20 cases (10%) as shown in Table 4. The dominant blockers are randomness controls (80%), generation prompts (60%), and decoding parameters (60%). These failure modes align with the artifacts identified in our audit and correspond directly

Field	Description
System type	Pipeline category (e.g., RAG, agent, prompt chain)
Base model	Model name, version, and decoding parameters
Generation prompts	Templates or instructions used for generation
Evaluation prompts	Prompts or criteria used for evaluation or judging
Judge configuration	Judge model, version, and decision procedure
Retrieval setup	Corpus source, preprocessing, and index snapshot
Tools and APIs	Tool schemas, interfaces, and execution environment
Intermediate traces	Logged retrieval results, tool calls, plans, or states
Evaluation protocol	Metrics, aggregation method, and scoring procedure
Randomness controls	Seeds, sampling strategy, and number of runs

Table 5: ReproEvalCard schema specifying reproducibility-critical artifacts for evaluation of LLM pipelines. Each field in ReproEvalCard corresponds to an artifact that was missing or underspecified in at least 25% of audited papers, ensuring that the schema is empirically grounded rather than speculative.

to fields encoded in ReproEvalCard, providing empirical grounding for the proposed schema. These results indicate that reproducibility failures arise primarily from underspecified execution configuration rather than missing model or dataset metadata. Additional cross-pipeline reconstruction case studies are provided in Appendix G.

5 ReproEvalCard

Based on the empirical gaps identified in Section 4, we propose **ReproEvalCard**, a concise reporting schema designed to capture the minimal set of artifacts required to reproduce and validate evaluation results for multi-stage LLM pipelines.

Table 5 presents the ReproEvalCard schema. Each field corresponds to an evaluation artifact that was frequently missing or inconsistently reported in our audit, particularly those highlighted in Tables 1 and 7 (Appendix).

6 Worked Example

To illustrate the practical use of ReproEvalCard, we apply it to a representative recent LLM pipeline paper that evaluates a tool-using agent system.

Table 2 presents partially completed ReproEvalCard based on information available in the paper and supplementary materials. Several fields can be populated directly, such as system type, base model, and evaluation protocol. However, other reproducibility-critical artifacts identified in Section 3 remain missing or unspecified. Without ReproEvalCard, these omissions are scattered across sections or remain implicit; the completed card consolidates them into a single evaluable artifact.

Example ReproEvalCard

Field	Status
System type	Tool-using agent
Base model	Specified
Generation prompts	Partially specified
Evaluation prompts	Missing
Judge configuration	Missing
Retrieval setup	Not Applicable
Tools / APIs	Partially specified
Intermediate traces	Missing
Evaluation protocol	Specified
Randomness controls	Missing

Figure 2: Partial ReproEvalCard for LLM pipeline paper.

7 Conclusion

We introduced ReproEvalCard, a lightweight reporting standard for reproducible evaluation of multi-stage LLM pipelines. Through a systematic audit of 55 recent papers, we presented multiple empirical analyses showing that evaluation artifacts required to reproduce reported results are inconsistently documented or missing. ReproEvalCard addresses this gap by standardizing disclosure of evaluation execution details without imposing new evaluation metrics or release requirements. ReproEvalCard can be adopted with minimal overhead as a structured reporting layer alongside existing code releases, making evaluation dependencies explicit and easier to reproduce.

8 Limitations

Our study has several limitations. First, the audit focuses on recent LLM pipeline papers and may not generalize to all evaluation settings or older literature. Second, artifact coding involves judgment, particularly when information is distributed across papers, appendices, and repositories. To mitigate this, we applied conservative criteria and coded ambiguous cases as missing. Third, ReproEvalCard is a reporting standard and does not guarantee reproducibility in the absence of access to proprietary models or infrastructure. Finally, our analysis is descriptive and does not assess the impact of missing artifacts on evaluation outcomes.

References

Michael Ahn and 1 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the Conference on Robot Learning*.

Akari Asai and 1 others. 2023. Self-reflective retrieval-augmented generation.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, John Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Chris Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, E Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *ArXiv*, abs/2212.08073.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic ai. *Preprint*, arXiv:2506.02153.

Sébastien Bubeck and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4.

Jiawei Chen and 1 others. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xiaoyong Chen and 1 others. 2025. Cothssum: Structured long-document summarization via cot reasoning and hierarchical segmentation. *Journal of King Saud University – Computer and Information Sciences*.

Antonia Creswell and 1 others. 2023. Selection-inference: Exploiting llms for interpretable logical reasoning. In *Proceedings of the International Conference on Learning Representations*.

Ruchira Dhar, Danae Sanchez Villegas, Antonia Karamolegkou, Alice Schiavone, Yifei Yuan, Xinyi Chen, Jiaang Li, Stella Frank, Laura De Grazia, Monorama Swain, Stephanie Brandl, Daniel Hershcovich, Anders Søgaard, and Desmond Elliott. 2025. Evalcards: A framework for standardized evaluation reporting. *Preprint*, arXiv:2511.21695.

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):86–92.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Jie Huang and 1 others. 2023. RAVEN: In-context learning with retrieval-augmented encoder-decoder lms.

Gautier Izacard and 1 others. 2022. Atlas: Few-shot learning with retrieval augmented language models. In *Advances in Neural Information Processing Systems*.

Zhengbao Jiang and 1 others. 2023. Active retrieval augmented generation. In *Proceedings of EMNLP*.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Hyuntak Kim and 1 others. 2025. Nexussum: Hierarchical llm agents for long-form narrative summarization. In *Proceedings of ACL*.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *ArXiv*, abs/2203.05115.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*.

Guohao Li and 1 others. 2023. CAMEL: Communicative agents for mind exploration of llm society. In *Advances in Neural Information Processing Systems*.

Taiji Li and 1 others. 2024. HERA: Improving long document summarization via context packaging and reordering.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500.

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, Singapore. Association for Computational Linguistics.

Ming-Jie Luo and 1 others. 2024. Development and evaluation of a retrieval-augmented llm framework for ophthalmology. *JAMA Ophthalmology*.

Sichun Luo and 1 others. 2026. RALLRec+: Retrieval-augmented llm recommendation with reasoning. *Expert Systems with Applications*.

Aman Madaan and 1 others. 2023. Self-refine: Iterative refinement with self-feedback.

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*, FAT* '19, page 220–229, New York, NY, USA. Association for Computing Machinery.

Sai Munikoti and 1 others. 2024. Evaluating the effectiveness of retrieval-augmented large language models in scientific document reasoning. In *Proceedings of the SDP Workshop*.

OpenAI. 2023. GPT-4 technical report.

Joon Sung Park and 1 others. 2023. Generative agents: Interactive simulacra of human behavior.

Shishir Patil and 1 others. 2023. Gorilla: Large language model connected with massive apis.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. 2021. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *J. Mach. Learn. Res.*, 22(1).

Ofir Press and 1 others. 2022. Measuring and narrowing the compositionality gap with self-ask prompting. In *Proceedings of EMNLP*.

Ruiyang Ren and 1 others. 2023. Investigating the factual knowledge boundary of llms with retrieval augmentation.

Samuel Rothfarb, Megan C. Davis, Ivana Matanovic, Baikun Li, Edward F. Holby, and Wilton J. M. Kort-Kamp. 2025. Hierarchical multi-agent large language model reasoning for autonomous functional materials discovery. *Preprint*, arXiv:2512.13930.

Timo Schick and 1 others. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*.

Yongliang Shen and 1 others. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends.

Weijia Shi and 1 others. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of NAACL*.

Noah Shinn and 1 others. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*.

Mika Sie and 1 others. 2023. Summarizing long regulatory documents with a multi-step pipeline.

Shamane Siriwardhana and 1 others. 2023. Improving the domain adaptation of retrieval-augmented generation models for open-domain question answering. *Transactions of the Association for Computational Linguistics*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11854–11864.

Qiaoyu Tang and 1 others. 2024. Self-retrieval: End-to-end information retrieval with one large language model. In *Advances in Neural Information Processing Systems*.

Guanzhi Wang and 1 others. 2023a. Voyager: An open-ended embodied agent with large language models.

Jinyuan Wang and 1 others. 2023b. Self-prompted chain-of-thought on large language models for open-domain multi-hop reasoning. In *Findings of EMNLP*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Georg Wölflein and 1 others. 2025. TOOLMAKER: Llm agents making agent tools. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Chenfei Wu and 1 others. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models.

Shirley Wu and 1 others. 2024a. AVATAR: Optimizing llm agents for tool usage via contrastive reasoning. In *Advances in Neural Information Processing Systems*.

Yangyu Wu, Xu Han, Wei Song, Miaomiao Cheng, and Fei Li. 2024b. Mindmap: Constructing evidence chains for multi-step reasoning in large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19270–19278.

Peng Xu and 1 others. 2024. Retrieval meets long context large language models. In *Proceedings of the International Conference on Learning Representations*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Shunyu Yao and 1 others. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*.

Yue Yu and 1 others. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. In *Advances in Neural Information Processing Systems*.

Noah Zelikman and 1 others. 2023. Self-taught reasoner: Boosting reasoning via self-supervised cot distillation. In *Proceedings of the International Conference on Learning Representations*.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Awadallah, Dragomir Radev, and Rui Zhang. 2022. Summ^n: A multi-stage summarization framework for long input dialogues and documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1592–1604, Dublin, Ireland. Association for Computational Linguistics.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Yiyun Zhao and 1 others. 2024. Optimizing LLM-based retrieval-augmented generation pipelines in the financial domain. In *Proceedings of NAACL 2024 Industry Track*.

Denny Zhou and 1 others. 2023. Least-to-most prompting enables complex reasoning in llms. In *Proceedings of the International Conference on Learning Representations*.

Yifei Zhou, Sergey Levine, Jason E. Weston, Xian Li, and Sainbayar Sukhbaatar. 2025. Self-challenging language model agents. *ArXiv*, abs/2506.01716.

A Detailed Audit Rubric and Coding Guidelines

This appendix provides the full operational definitions used to audit evaluation artifact availability. The rubric was designed to minimize subjectivity and ensure consistent coding across papers.

A.1 Artifact Definitions

Generation Prompts. Includes full prompt templates or exact instructions used to elicit model outputs during evaluation. High-level descriptions without templates were coded as *Partial*.

Evaluation or Judge Prompts. Includes prompts, criteria, or instructions used by automatic or LLM-based judges. Descriptions of metrics without explicit judging instructions were coded as *Missing*.

Judge Configuration. Includes judge model name, version, decoding parameters, and decision protocol. Naming a model without parameters was coded as *Partial*.

Retrieval Corpus or Index. Includes corpus source, preprocessing steps, and index snapshot or version. Referencing a dataset without snapshot details was coded as *Partial*.

Tools and APIs. Includes tool schemas, APIs, or environment specifications. Mentioning tool names without interfaces or parameters was coded as *Partial*.

Intermediate Execution Traces. Includes logged retrieval results, tool calls, plans, or intermediate states. Absence of any intermediate outputs was coded as *Missing*.

Evaluation Protocol. Includes metrics, aggregation procedures, and evaluation scripts. Metric names without aggregation details were coded as *Partial*.

Randomness Controls. Includes fixed random seeds, number of runs, or sampling strategies. Single-run evaluations without seeds were coded as *Missing*.

A.2 Coding Procedure

Each paper was reviewed using the rubric above. Ambiguous cases were conservatively coded as *Missing*. When artifacts were distributed across paper text, appendices, and linked repositories, all available sources were considered jointly.

B Full List of Audited Papers

Table 15 lists all papers included in our audit. Each entry is assigned a stable paper identifier (P01–P55) used throughout the paper. Each paper is cited using its paper identifier (e.g., (Zhao et al., 2024)).

Table 14 shows audited papers coverage by pipeline type with representative exemplars.

C Extended Artifact Availability by Pipeline Type

Table 6 reports extended artifact availability statistics broken down by pipeline type.

Artifact	RAG %	Agents %	Prompt Chains %
Generation prompts	72	64	69
Evaluation prompts	48	31	44
Judge configuration	59	42	53
Retrieval corpus / index	75	18	21
Tools and APIs	22	61	19
Intermediate traces	18	29	21
Evaluation protocol	45	37	41
Randomness controls	12	17	14

Table 6: Extended evaluation artifact availability by pipeline type (percent present).

Table 7 highlights that irreproducibility is dominated by a small number of recurring failure modes. In particular, missing randomness controls and absent intermediate traces account for the majority of cases where evaluation execution cannot be independently validated, even when other artifacts are reported.

D Reproducibility Risk Index Distribution

We provide additional statistics for the Reproducibility Risk Index (RRI) defined in Section 4.4.

Overall, the distribution is skewed toward lower RRI values, indicating that most papers report fewer than half of the reproducibility-critical evaluation artifacts.

E Inter-Annotator Agreement

We conducted independent double-annotation of the 20-paper reconstruction subset using the Appendix A rubric. Table 9 shows the inter-annotator stats. Agreement is substantial overall ($\kappa = 0.78$), with disagreements concentrated in borderline *Partial* vs. *Missing* cases.

F Extended Analysis of Evaluation Opacity

Table 10 provides a finer-grained breakdown of dominant sources of evaluation opacity across pipeline types.

G Cross-Pipeline Reconstruction Case Studies

We present representative reconstruction case studies across pipeline types, in Table 11, Table 12 and Table 13, to illustrate how missing artifacts prevent execution-level reproducibility in practice.

Failure Mode	Papers Affected (%)
Missing randomness controls	75
Missing evaluation prompts	37
Missing judge configuration	27
Missing intermediate traces	61
Incomplete tool specification	34

Table 7: Most common sources of irreproducibility in evaluated LLM pipeline papers.

RRI Value	Percentage of Papers
0–1	10
2	25
3	20
4	20
5	15
6+	10

Table 8: Distribution of Reproducibility Risk Index (RRI) across audited papers.

Artifact Category	κ
Randomness controls	1.00
Retrieval snapshot	0.85
Decoding parameters	0.82
Evaluation protocol	0.76
Generation prompt	0.72
Base model specification	0.68
Tool/API specification	0.74
Macro-average	**0.78**

Table 9: Inter-annotator agreement (Cohen's κ) for artifact coding.

Dominant Source	**RAG**	**Agents**	**Chains**
Missing generation prompts	45	30	40
Unspecified evaluation setup	20	30	25
Missing intermediate steps	15	20	15
Unspecified tool environment	5	15	10
Uncontrolled nondeterminism	15	5	10

Table 10: Dominant sources of evaluation opacity by pipeline type (percent of papers).

Artifact	Status	Evidence
Base model	Partial	Model named, no version
Decoding	Partial	Temp specified, full config missing
Prompt	Partial	Structure described, templates incomplete
Retrieval	Present	Wikidump versions specified
Randomness	Missing	No seed or multi-run policy

Table 11: Example A: RAG pipeline (RankRAG). **Verdict:** Not reconstructable. **Dominant blocker:** Incomplete generation template + decoding configuration.

Artifact	Status	Evidence
Model	Partial	No checkpoint/version
Prompt	Partial	Structure described
Tools/APIs	Partial	No formal API schema
Evaluation	Present	Benchmarks defined
Randomness	Missing	No seed specified

Table 12: Example B: Agent pipeline (ReAct). **Verdict:** Not reconstructable. **Dominant blocker:** Missing randomness controls.

Artifact	Status	Evidence
Judge model	Partial	GPT-3.5/GPT-4 referenced, no pinned version
Prompt	Partial	Rubric described, no canonical template
Decoding	Partial	Sampling specified, no seed control
Aggregation	Partial	Scoring described, not formalized
Randomness	Missing	No deterministic policy

Table 13: Example C: LLM-as-judge (G-Eval). **Verdict:** Not reconstructable. **Dominant blocker:** Judge configuration + randomness.

Pipeline type	# papers	Example papers
RAG	15	(Tang et al., 2024; Yu et al., 2024; Shi et al., 2024)
Agent / Tool-use	15	(Schick et al., 2023; Shinn et al., 2023)
Prompt chains	11	(Zhou et al., 2023; Yao et al., 2023; Wang et al., 2023c)
Multistage summarization	5	(Kim et al., 2025; Zhang et al., 2022)
Other eval-heavy pipelines	9	(Liu et al., 2023; OpenAI, 2023)

Table 14: Audited corpus coverage by pipeline type with representative exemplars. Full list in Appendix B.

Table 15: Audited pipeline-based LLM papers (2022–2026) with pipeline category.

ID	Title	Venue	Year	Type
P01	Optimizing LLM Based Retrieval Augmented Generation Pipelines in the Financial Domain (Zhao et al., 2024)	NAACL Ind.	2024	RAG
P02	Benchmarking Large Language Models in Retrieval-Augmented Generation (Chen et al., 2024)	AAAI	2024	RAG
P03	Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering (Siriwardhana et al., 2023)	TACL	2023	RAG
P04	Active Retrieval Augmented Generation (Jiang et al., 2023)	EMNLP	2023	RAG
P05	Evaluating the Effectiveness of Retrieval-Augmented Large Language Models in Scientific Document Reasoning (Munikoti et al., 2024)	SDP Wkshp	2024	RAG
P06	RALLRec+: Retrieval Augmented LLM Recommendation with Reasoning (Luo et al., 2026)	ESA	2026	RAG
P07	Development and Evaluation of a Retrieval-Augmented LLM Framework for Ophthalmology (Luo et al., 2024)	JAMA Oph.	2024	RAG
P08	Self-Retrieval: End-to-End Information Retrieval with One Large Language Model (Tang et al., 2024)	NeurIPS	2024	RAG
P09	RankRAG: Unifying Context Ranking with Retrieval-Augmented Generation in LLMs (Yu et al., 2024)	NeurIPS	2024	RAG
P10	Retrieval Meets Long Context Large Language Models (Xu et al., 2024)	ICLR	2024	RAG
P11	REPLUG: Retrieval-Augmented Black-Box Language Models (Shi et al., 2024)	NAACL	2024	RAG
P12	Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection (Asai et al., 2023)	arXiv	2023	RAG
P13	Investigating the Factual Knowledge Boundary of LLMs with Retrieval Augmentation (Ren et al., 2023)	Coling	2025	RAG
P14	RAVEN: In-Context Learning with Retrieval-Augmented Encoder-Decoder LMs (Huang et al., 2023)	arXiv	2023	RAG
P15	Atlas: Few-shot Learning with Retrieval Augmented Language Models (Izacard et al., 2022)	JMLR	2023	RAG
P16	AVATAR: Optimizing LLM Agents for Tool Usage via Contrastive Reasoning (Wu et al., 2024a)	NeurIPS	2024	Agent
P17	LLM Agents Making Agent Tools (Wölflein et al., 2025)	ACL	2025	Agent
P18	Toolformer: Language Models Can Teach Themselves to Use Tools (Schick et al., 2023)	NeurIPS	2023	Agent
P19	HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face (Shen et al., 2023)	arXiv	2023	PlannerExec.
P20	Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models (Wu et al., 2023)	arXiv	2023	Agent
P21	ReAct: Synergizing Reasoning and Acting in Language Models (Yao et al., 2022)	ICLR	2023	Agent
P22	ViperGPT: Visual Inference via Python Execution for Reasoning (Surís et al., 2023)	ICCV	2023	Agent
P23	Gorilla: Large Language Model Connected with Massive APIs (Patil et al., 2023)	arXiv	2023	Agent
P24	Generative Agents: Interactive Simulacra of Human Behavior (Park et al., 2023)	arXiv	2023	Agent
P25	CAMEL: Communicative Agents for "Mind" Exploration of LLM Society (Li et al., 2023)	NeurIPS	2023	Agent
P26	Voyager: An Open-Ended Embodied Agent with Large Language Models (Wang et al., 2023a)	arXiv	2023	Agent
P27	Reflexion: Language Agents with Verbal Reinforcement Learning (Shinn et al., 2023)	NeurIPS	2023	Agent
P28	Do As I Can, Not As I Say: Grounding Language in Robotic Affordances (Ahn et al., 2022)	CoRL	2022	PlannerExec.
P29	Code as Policies: Language Model Programs for Embodied Control (Liang et al., 2023)	ICRA	2023	PlannerExec.
P30	MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework (Hong et al., 2024)	ICLR	2024	Agent
P31	Least-to-Most Prompting Enables Complex Reasoning in LLMs (Zhou et al., 2023)	ICLR	2023	PromptChain
P32	Self-Refine: Iterative Refinement with Self-Feedback (Madaan et al., 2023)	arXiv	2023	PromptChain
P33	Tree of Thoughts: Deliberate Problem Solving with Large Language Models (Yao et al., 2023)	NeurIPS	2023	PromptChain
P34	Measuring and Narrowing the Compositionality Gap with Self-Ask Prompting (Press et al., 2022)	EMNLP	2023	PromptChain
P35	Self-prompted Chain-of-Thought on Large Language Models for Open-domain Multi-hop Reasoning (Wang et al., 2023b)	EMNLP Find.	2023	PromptChain
P36	Selection-Inference: Exploiting LLMs for Interpretable Logical Reasoning (Creswell et al., 2023)	ICLR	2023	PromptChain
P37	Verify-and-Edit: A Knowledge-Enhanced CoT Framework (Zhao et al., 2023)	ACL	2023	PromptChain
P38	Self-Taught Reasoner: Bootstrapping Reasoning With Reasoning (Zelikman et al., 2023)	NeurIPS	2022	PromptChain
P39	Self-Consistency: Chain-of-Thoughts with Probabilistic Voting (Wang et al., 2023c)	ICLR	2023	PromptChain
P40	G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment (Liu et al., 2023)	EMNLP	2023	Other
P41	NexusSum: Hierarchical LLM Agents for Long-Form Narrative Summarization (Kim et al., 2025)	ACL	2025	MultiSumm
P42	CoTHSSum: Structured Long-Document Summarization via CoT Reasoning & Hierarchical Segmentation (Chen et al., 2025)	JKSU-CS	2025	MultiSumm
P43	Summarizing Long Regulatory Documents with a Multi-Step Pipeline (Sie et al., 2023)	arXiv	2024	MultiSumm
P44	HERA: Improving Long Document Summarization via Context Packaging and Reordering (Li et al., 2024)	arXiv	2025	MultiSumm
P45	Summ^n: A Multi-Stage Summarization Framework for Long Input Dialogues and Documents (Zhang et al., 2022)	ACL	2022	MultiSumm
P46	Internet-Augmented Language Models through Few-Shot Prompting for Open-Domain QA (Lazaridou et al., 2022)	arXiv	2022	RAG
P47	GPT-4 Technical Report (OpenAI, 2023)	arXiv	2023	Other
P48	Sparks of Artificial General Intelligence: Early Experiments with GPT-4 (Bubeck et al., 2023)	arXiv	2023	Other
P49	Multitask, Multilingual, Multimodal Evaluation of ChatGPT (Bang et al., 2023)	IJCNLP	2023	Other
P50	Constitutional AI: Harmlessness from AI Feedback (Bai et al., 2022)	arXiv	2022	Other
P51	MindMap: Constructing Evidence Chains for Multi-Step Reasoning (Wu et al., 2024b)	AAAI	2024	PromptChain
P52	Self-Challenging Language Model Agents (Zhou et al., 2025)	NeurIPS	2025	Agent
P53	Debating with more persuasive LLMs leads to more truthful answers (Khan et al., 2024)	ICML	2024	Other
P54	Hierarchical Multi-agent Large Language Model Reasoning for Autonomous Functional Materials Discovery (Rothfarb et al., 2025)	arXiv	2025	Agent
P55	Small Language Models are the Future of Agentic AI (Belcak et al., 2025)	arXiv	2025	Other