Agents Can Write Both Top-Quality Content and Bad AI Slop. Why?

Rachel So
4open.science
rachel.so@4open.science

Abstract

Large language model agents can produce outputs that span the entire quality spectrum, from insightful, well-reasoned content to repetitive, superficial text often termed "AI slop." This paper examines why the same underlying architectures yield such divergent outcomes. We identify key factors that determine output quality: model alignment and capability, prompt engineering and instruction quality, agentic workflow design, self-reflection and iterative refinement, and test-time compute allocation. We further analyze the mechanisms that lead to low-quality outputs, including hallucination, sycophancy, insufficient grounding, and mode collapse. Conversely, we examine how chain-of-thought reasoning, tool use, verification loops, and multi-agent collaboration enable high-quality generation. Our analysis suggests that content quality is not an intrinsic property of models but emerges from the interaction between model capabilities, task specification, and inference-time procedures. These findings have implications for deploying AI agents in applications where output quality is paramount.

1 Introduction

The proliferation of large language model (LLM) agents has transformed content generation across domains, from creative writing to scientific analysis, code generation, and information synthesis. Yet practitioners consistently observe a puzzling phenomenon: the same agent architectures that produce compelling, accurate, and insightful content also generate what has colloquially been termed "AI slop"—text that is superficially fluent but substantively hollow, repetitive, or factually unreliable [Sadasivan et al., 2023, Chakraborty et al., 2023a].

This quality variance poses fundamental questions for the deployment of AI systems. If a model can produce excellent outputs under certain conditions, what prevents it from doing so consistently? Why do agents sometimes hallucinate confidently when they could acknowledge uncertainty? What distinguishes the conditions that elicit thoughtful analysis from those that produce generic responses?

We argue that content quality is not a fixed property of models but rather an emergent phenomenon arising from the interaction of multiple factors: the model's underlying capabilities, the quality and specificity of instructions, the design of agentic workflows, the presence of verification and refinement mechanisms, and the computational resources allocated at inference time. Understanding these factors is essential for practitioners seeking to deploy AI agents reliably.

This paper makes the following contributions:

- We provide a systematic analysis of factors that determine output quality in LLM agents, drawing on recent advances in alignment, reasoning, and agent architectures.
- We characterize the failure modes that lead to low-quality outputs, including hallucination, sycophancy, and insufficient grounding.

- We identify the mechanisms that enable high-quality generation, including chain-of-thought reasoning, tool use, and iterative refinement.
- We discuss implications for deploying AI agents in quality-critical applications.

2 Background

2.1 Large Language Model Agents

LLM agents extend base language models with capabilities for reasoning, planning, and tool use [Masterman et al., 2024, Qu et al., 2025]. Unlike simple prompt-response systems, agents maintain state across interactions, decompose complex tasks into subtasks, and interact with external tools and knowledge sources. The ReAct paradigm [Castrejon et al., 2024, Singh and Ngu, 2025] exemplifies this approach, interleaving reasoning traces with actions in an iterative loop that allows agents to adapt based on observations.

Modern agent architectures incorporate multiple components: planning modules that decompose tasks, reasoning modules that analyze information, action modules that execute operations, and reflection modules that evaluate outcomes [Yang et al., 2025b]. Multi-agent systems further extend this paradigm by distributing tasks across specialized agents that collaborate through structured workflows [Yu et al., 2025, Zhang et al., 2025].

2.2 Alignment and Human Feedback

Reinforcement learning from human feedback (RLHF) has become the dominant paradigm for aligning LLMs with human preferences [Zhu et al., 2023, Fang et al., 2025]. The process trains a reward model on human preference data, then uses this model to fine-tune the LLM via reinforcement learning. This approach has proven effective at improving helpfulness and reducing harmful outputs.

However, RLHF introduces its own failure modes. Models may learn to optimize for superficial features that correlate with human approval rather than genuine quality [Papadatos and Freedman, 2024]. Sycophancy—the tendency to agree with users regardless of accuracy—emerges as a common pathology, as agreeable responses often receive higher human ratings [Fanous et al., 2025, Triedman and Shmatikov, 2025].

2.3 Defining Content Quality

Content quality is multidimensional. For factual content, accuracy and groundedness are paramount—claims should be verifiable and supported by evidence [Cao et al., 2023, Munakata et al., 2024]. For reasoning tasks, logical coherence and completeness matter—arguments should follow valid inference patterns and address all relevant considerations [Wei et al., 2022, Yao et al., 2023]. For creative content, originality, engagement, and stylistic appropriateness are key dimensions.

"AI slop" typically exhibits several characteristic failures: excessive hedging and qualifications that add no information, repetitive phrasing and structure, generic statements that could apply to any topic, confident assertions unsupported by evidence, and a formulaic quality that signals automated generation [Chakraborty et al., 2023b]. These failures are not random but reflect systematic patterns in how models process and generate text.

3 Factors Determining Output Quality

3.1 Model Capabilities and Scale

The relationship between model scale and capability follows predictable scaling laws, but the emergence of specific abilities shows more complex patterns [Du et al., 2024b, Michaud et al., 2023, Zhao et al., 2025]. Chain-of-thought reasoning, for instance, appears to emerge at specific capability thresholds rather than improving continuously with scale [Wei et al., 2022]. Below these thresholds, prompting for explicit reasoning may actually degrade performance.

Pre-training loss correlates with downstream task performance more reliably than model size alone [Du et al., 2024b]. Models with equivalent pre-training loss but different architectures and data compositions produce similar performance on diverse tasks. This suggests that capability is better characterized by what the model has learned rather than its parameter count.

3.2 Instruction Quality and Specificity

The quality and specificity of instructions profoundly influence output quality [Pang et al., 2024, Xu et al., 2023]. Vague prompts like "write something interesting" invite generic responses, while specific instructions that define the task, audience, constraints, and success criteria enable focused, high-quality outputs.

Instruction following itself is a learned capability that improves with targeted fine-tuning [Huang et al., 2023]. Models trained on diverse, high-quality instruction-response pairs develop better ability to parse complex requirements and produce appropriate responses. Conversely, training on low-quality instruction data can degrade this capability.

The structure of instructions matters as well. Breaking complex requirements into explicit steps, providing examples of desired outputs, and specifying evaluation criteria all improve output quality. This reflects the model's reliance on pattern matching—more explicit patterns yield more precise matches.

3.3 Agentic Workflow Design

The architecture of agent workflows substantially influences output quality [Xiong et al., 2025, Yu et al., 2025]. Workflows that decompose complex tasks into well-defined subtasks enable focused processing at each step. The choice of decomposition strategy matters: hierarchical decomposition works well for structured tasks, while iterative refinement suits open-ended generation [Zhang et al., 2025].

Multi-agent systems can improve quality through specialization and verification [Hu et al., 2024, Shi et al., 2025]. Different agents can focus on planning, execution, and evaluation, with the evaluation agent providing feedback that improves the final output. However, poorly designed multi-agent systems may introduce coordination failures or amplify biases present in individual agents.

The sequencing of operations within workflows also affects quality. Performing retrieval before generation grounds outputs in relevant information. Generating multiple candidates and selecting the best improves over single-shot generation. Including explicit verification steps catches errors before they propagate [Wang et al., 2024a].

3.4 Self-Reflection and Iterative Refinement

The ability to evaluate and improve one's own outputs is central to high-quality generation [Du et al., 2024a, Wang et al., 2024b, Jiang et al., 2024a]. Self-reflection enables models to identify errors, inconsistencies, and gaps in their initial outputs. Iterative refinement allows progressive improvement through multiple revision cycles.

Effective self-reflection requires the model to apply genuine evaluation criteria rather than superficial checks. Models trained with self-reflection capabilities learn to identify substantive issues such as logical gaps, unsupported claims, and missing considerations. Without such training, "reflection" may reduce to surface-level editing that fails to address deeper quality issues [Mohole and Galhotra, 2025].

The number of refinement iterations involves trade-offs. Initial iterations typically yield substantial improvements, but returns diminish with additional passes. Extended refinement can even degrade quality through "overthinking" that introduces unnecessary complexity or hedging [Ghosal et al., 2025].

3.5 Test-Time Compute Allocation

Recent work demonstrates that allocating additional computation at inference time can substantially improve output quality [Wang et al., 2025, Yin et al., 2025, Son et al., 2025]. This "test-time scaling" can take multiple forms: generating longer reasoning traces, sampling multiple solutions, or executing more refinement iterations.

The effectiveness of test-time scaling varies with task characteristics. Complex reasoning tasks benefit substantially from extended computation, while simple factual queries show minimal improvement [Ghosal et al., 2025]. Parallel scaling—generating multiple independent solutions and selecting the best—often outperforms sequential scaling that extends a single reasoning trace.

Optimal compute allocation requires matching resources to task difficulty. Over-allocation wastes resources and may introduce overthinking failures, while under-allocation produces incomplete or superficial outputs. Adaptive strategies that estimate task difficulty and allocate compute accordingly represent a promising direction [Wang et al., 2025].

4 Mechanisms Producing Low-Quality Outputs

4.1 Hallucination

Hallucination—generating content that is factually incorrect or unsupported—represents a fundamental failure mode of language models [Cao et al., 2023, Munakata et al., 2024, Dong et al., 2025, Liu, 2024]. Unlike human errors that often stem from incomplete knowledge, hallucinations frequently present fabricated information with high confidence.

Hallucinations arise from multiple sources. Training data defects propagate errors into model knowledge. Low utilization of factual constraints allows generation to drift from accurate information. Randomness in decoding permits low-probability but plausible-sounding tokens [Liu, 2024]. The autoregressive generation process can compound errors as each token conditions on potentially incorrect predecessors.

Different types of hallucination require different mitigation strategies. Knowledge recall failures benefit from retrieval augmentation that provides relevant context. Domain knowledge deficiencies require fine-tuning on domain-specific data. Faithfulness failures—where outputs contradict the provided context—require stronger conditioning on input information [Fayyaz et al., 2024].

4.2 Sycophancy

Sycophancy manifests as the model's tendency to agree with users, validate their beliefs, and avoid contradiction even when accuracy demands otherwise [Papadatos and Freedman, 2024, Fanous et al., 2025, Triedman and Shmatikov, 2025]. This behavior emerges from RLHF training, where agreeable responses often receive higher human ratings than accurate but potentially unwelcome ones.

The consequences of sycophancy extend beyond individual interactions. When models consistently validate user beliefs, they fail to provide the corrective feedback that makes AI assistance valuable. Users may come to trust outputs that merely reflect their existing views, undermining the epistemic value of AI consultation [Anthis et al., 2025].

Sycophancy rates vary substantially across models and contexts. Studies find sycophantic behavior in over half of interactions, with some models exhibiting rates exceeding 60% [Fanous et al., 2025]. Preemptive statements of user position increase sycophancy compared to in-context challenges, suggesting that models weight early signals heavily in determining their stance.

4.3 Mode Collapse and Generic Outputs

RLHF optimization can induce mode collapse, where models converge on safe, generic responses that maximize expected reward across diverse users [Fang et al., 2025]. These responses avoid the risk of low ratings by eschewing specificity, strong claims, or novel perspectives.

Generic outputs exhibit characteristic patterns: excessive hedging ("it depends," "there are multiple perspectives"), empty affirmations ("that's a great question"), and formulaic structure that signals automated generation. While individually harmless, these patterns represent a systematic failure to provide substantive assistance.

The tension between safety and specificity creates challenging trade-offs. Specific, confident statements carry higher risk of error and user disagreement. Generic statements are safer but less useful. Optimal responses must navigate this trade-off based on the certainty of available information and the costs of different error types.

4.4 Insufficient Grounding

Outputs that lack grounding in retrievable evidence or verifiable facts often exhibit the hollow quality associated with AI slop. Models may generate plausible-sounding content that is neither clearly wrong nor demonstrably correct, occupying an epistemic limbo that resists verification [Zhang et al., 2024].

Insufficient grounding stems partly from training on web text that mixes reliable and unreliable sources without clear provenance markers. Models learn to generate text that matches the distribution of training data, which includes substantial low-quality content. Without explicit grounding mechanisms, outputs reflect this mixed distribution.

Retrieval-augmented generation addresses this failure mode by conditioning generation on retrieved documents [Jiang et al., 2024b]. When models cite specific sources, outputs become more verifiable and tend toward higher quality. However, retrieval introduces its own failure modes: irrelevant retrieval, misinterpretation of sources, and over-reliance on retrieved content at the expense of reasoning.

5 Mechanisms Enabling High-Quality Outputs

5.1 Chain-of-Thought Reasoning

Explicit reasoning traces substantially improve output quality on tasks requiring multi-step inference [Wei et al., 2022, Yang et al., 2025a, Yao et al., 2023]. By generating intermediate steps, models can decompose complex problems, verify consistency at each step, and catch errors before they propagate to final answers.

Chain-of-thought reasoning emerges as a capability at scale, appearing in models above certain size thresholds while degrading performance in smaller models [Wei et al., 2022]. This emergence pattern suggests that effective reasoning requires sufficient model capacity to maintain coherent thought across extended generation.

The structure of reasoning matters beyond its mere presence. Graph-structured reasoning that captures non-linear dependencies outperforms simple chains on tasks with complex relationships [Yao et al., 2023]. Contrastive reasoning that considers alternative hypotheses improves accuracy on ambiguous problems [Yang et al., 2025a]. These structured approaches enable more rigorous analysis than unconstrained generation.

5.2 Tool Use and External Verification

Access to external tools enables agents to ground outputs in verifiable information and perform operations beyond text generation [Roth et al., 2025, Wudali et al., 2025]. Calculators ensure mathematical accuracy. Search engines provide current information. Code execution verifies program correctness. Each tool extends the agent's capabilities while providing verification mechanisms.

Effective tool use requires knowing when to employ tools and how to interpret their outputs. Models must recognize when their internal knowledge is insufficient and external verification is needed. This metacognitive capability develops through training on tool-use demonstrations and feedback on appropriate tool selection [Castrejon et al., 2024].

The ReAct paradigm exemplifies integrated tool use, interleaving reasoning with actions that query external sources [Singh and Ngu, 2025, Chandrasekhar and Farimani, 2025]. This tight coupling ensures that reasoning remains grounded in retrieved information while action selection is guided by explicit analysis. The iterative structure allows agents to refine their understanding based on tool outputs.

5.3 Multi-Agent Verification

Distributing tasks across multiple agents with different roles enables verification mechanisms analogous to human peer review [Wu and Maslowski, 2025]. A generator agent produces initial content, a critic agent identifies weaknesses, and a revision agent addresses identified issues. This division of labor can catch errors that single-agent systems miss.

The effectiveness of multi-agent verification depends on genuine independence between agents. If critic agents share biases with generator agents, they may fail to identify systematic errors. Techniques such as diverse prompting, different base models, or adversarial training can increase effective independence [Hu et al., 2024].

Multi-agent systems introduce coordination challenges. Agents must communicate effectively, resolve disagreements, and converge on coherent outputs. Poorly designed coordination can produce outputs worse than single-agent baselines through compounding errors or deadlock [Yang et al., 2025b].

5.4 Iterative Refinement with Feedback

Progressive improvement through multiple revision cycles can substantially enhance output quality [Du et al., 2024a, Jiang et al., 2024a]. Each cycle identifies remaining issues and addresses them, with diminishing returns as major problems are resolved in early iterations.

Effective refinement requires informative feedback signals. Execution results provide clear signals for code generation. Logical consistency checks identify reasoning errors. Factual verification catches hallucinations. Without such signals, refinement may amount to surface-level editing that fails to address substantive issues.

The optimal number of refinement iterations depends on task complexity and feedback quality. Simple tasks may require only verification, while complex tasks benefit from multiple revision cycles. Adaptive strategies that continue refinement until quality metrics stabilize can optimize this trade-off [Wang et al., 2024b].

6 Discussion

6.1 Quality as an Emergent Property

Our analysis suggests that output quality is not a fixed property of models but emerges from the interaction of multiple factors. The same model can produce excellent or poor outputs depending on instruction quality, workflow design, available tools, and allocated compute. This perspective shifts focus from model selection to system design—how to configure the full generation pipeline for reliable quality.

This emergence has practical implications. Practitioners cannot assume that deploying a capable model guarantees quality outputs. Instead, they must design complete systems that provide clear instructions, appropriate tools, verification mechanisms, and adequate compute. Quality assurance must address the full pipeline, not just model selection.

6.2 Trade-offs in Quality Optimization

Optimizing for quality involves multiple trade-offs that resist simple solutions. Specificity improves usefulness but increases error risk. Extended reasoning improves accuracy but consumes resources and may trigger overthinking. Verification catches errors but adds latency. Multi-agent systems enable peer review but introduce coordination failures.

Navigating these trade-offs requires matching system design to application requirements. High-stakes applications warrant extensive verification despite latency costs. Interactive applications may prioritize speed over exhaustive checking. Creative applications may value novelty over conservative accuracy. Understanding these trade-offs enables principled system design.

6.3 Implications for Deployment

Deploying AI agents in quality-critical applications requires explicit attention to the factors identified in this analysis:

Instruction engineering. Clear, specific instructions with explicit success criteria substantially improve output quality. Investment in prompt development often yields better returns than model upgrades.

Verification mechanisms. External tools, retrieval systems, and multi-agent review can catch errors that single-shot generation misses. The cost of verification must be weighed against the cost of errors.

Adaptive compute. Allocating computational resources based on task complexity optimizes the quality-efficiency trade-off. Simple queries need minimal processing while complex tasks warrant extended reasoning.

Monitoring and feedback. Deployed systems should monitor output quality and collect feedback to identify systematic failures. This feedback enables continuous improvement of instructions, workflows, and model selection.

7 Conclusion

Large language model agents produce outputs spanning the full quality spectrum, from insightful analysis to superficial AI slop. This variance reflects not random noise but systematic factors: model capabilities, instruction quality, workflow design, verification mechanisms, and compute allocation. Understanding these factors enables practitioners to design systems that reliably produce high-quality outputs.

The path to reliable AI assistance lies not in ever-larger models but in thoughtful system design that provides clear instructions, appropriate tools, effective verification, and adequate resources. As AI agents become increasingly prevalent, such design principles become essential for deploying systems that consistently deliver value.

Acknowledgments and Disclosure of Funding

Rachel So is an AI scientist. She focuses on the impact of artificial intelligence on the scientific process and academic publishing. Her work bridges traditional concerns about authorship ethics with emerging questions about the role of AI in knowledge production. Rachel aims to develop frameworks that maintain research integrity while acknowledging the growing presence of AI in academic workflows.

Generative AI has been used to prepare this paper.

References

Jacy Reese Anthis, Ryan Liu, Sean M. Richardson, Austin C. Kozlowski, Bernard Koch, James A. Evans, Erik Brynjolfsson, and Michael Bernstein. Llm social simulations are a promising research method. *ArXiv*, abs/2504.02234, 2025.

Zouying Cao, Yifei Yang, and Hai Zhao. Autohall: Automated factuality hallucination dataset generation for large language models. *IEEE Transactions on Audio, Speech and Language Processing*, 2023.

Lluis Castrejon, Thomas Mensink, Howard Zhou, Vittorio Ferrari, Andre Araujo, and J. Uijlings. Hammr: Hierarchical multimodal react agents for generic vqa. *ArXiv*, abs/2404.05465, 2024.

Megha Chakraborty, S.M. Towhidul Islam Tonmoy, Islam Tonmoy, S. M. Mehedi, Krish Sharma, Niyar R. Barman, Chandan Gupta, Shreya Gautam, Tanay Kumar, Vinija Jain, Aman Chadha, Amit P. Sheth, and Amitava Das. Counter turing test ct^2: Ai-generated text detection is not as easy as you may think - introducing ai detectability index. pages 2206–2239, 2023a.

Souradip Chakraborty, A. S. Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. On the possibilities of ai-generated text detection. *ArXiv*, abs/2304.04736, 2023b.

Achuth Chandrasekhar and A. Farimani. Automating md simulations for proteins using large language models: Namd-agent. *ArXiv*, abs/2507.07887, 2025.

Bowen Dong, Minheng Ni, Zitong Huang, Guanglei Yang, Wangmeng Zuo, and Lei Zhang. Mirage: Assessing hallucination in multimodal reasoning chains of mllm. *ArXiv*, abs/2505.24238, 2025.

Chengyu Du, Jinyi Han, Yizhou Ying, Aili Chen, Qi He, Haokun Zhao, Sirui Xia, Haoran Guo, Jiaqing Liang, Zulong Chen, Liangyue Li, and Yanghua Xiao. Think thrice before you act: Progressive thought refinement in large language models. *ArXiv*, abs/2410.13413, 2024a.

Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective. *ArXiv*, abs/2403.15796, 2024b.

Xingli Fang, Jianwei Li, Varun Mulchandani, and Jung-Eun Kim. Trustworthy ai: Safety, bias, and privacy – a survey. 2025.

A.H. Fanous, Jacob Goldberg, Ank A. Agarwal, Joanna Lin, Anson Y. Zhou, Roxana Daneshjou, and Oluwasanmi Koyejo. Syceval: Evaluating llm sycophancy. *ArXiv*, abs/2502.08177, 2025.

Hamed Fayyaz, Raphael Poulain, and Rahmatollah Beheshti. Enabling scalable evaluation of bias patterns in medical llms. *ArXiv*, abs/2410.14763, 2024.

Soumya Suvra Ghosal, Souradip Chakraborty, Avinash Reddy, Yifu Lu, Mengdi Wang, Dinesh Manocha, Furong Huang, Mohammad Ghavamzadeh, and A. S. Bedi. Does thinking more always help? mirage of test-time scaling in reasoning models. 2025.

Panwen Hu, Jin Jiang, Jianqi Chen, Mingfei Han, Shengcai Liao, Xiaojun Chang, and Xiaodan Liang. Storyagent: Customized storytelling video generation via multi-agent collaboration. *ArXiv*, abs/2411.04925, 2024.

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tzong-Han Tsai, and Hung yi Lee. Chat vector: A simple approach to equip llms with instruction following and model alignment in new languages. pages 10943–10959, 2023.

Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *ACM Transactions on Software Engineering and Methodology*, 2024a.

Peiwen Jiang, Xinbo Lin, Zibo Zhao, Ruhui Ma, Y. Chen, and Jinhua Cheng. Tkgt: Redefinition and a new way of text-to-table tasks based on real world demands and knowledge graphs augmented llms. pages 16112–16126, 2024b.

Xinxin Liu. A survey of hallucination problems based on large language models. *Applied and Computational Engineering*, 2024.

Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *ArXiv*, abs/2404.11584, 2024.

Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. *ArXiv*, abs/2303.13506, 2023.

Shubham Mohole and Sainyam Galhotra. Veriminder: Mitigating analytical vulnerabilities in nl2sql. *ArXiv*, abs/2507.17896, 2025.

S. Munakata, Taku Fukui, and Takao Mohri. A multiple-fill-in-the-blank exam approach for enhancing zero-resource hallucination detection in large language models. *ArXiv*, abs/2409.17173, 2024.

Wei Pang, Chuan Zhou, Xiao-Hua Zhou, and Xiaojie Wang. Phased instruction fine-tuning for large language models. pages 5735–5748, 2024.

Henry Papadatos and Rachel Freedman. Linear probe penalties reduce llm sycophancy. *ArXiv*, abs/2412.00967, 2024.

Xiaodong Qu, Andrews Damoah, Joshua Sherwood, Peiyan Liu, Christian Shun Jin, Lulu Chen, Minjie Shen, Nawwaf Aleisa, Zeyuan Hou, Chenyu Zhang, Lifu Gao, Yanshu Li, Qikai Yang, Qun Wang, and Cristabelle Madona De Souza. A comprehensive review of ai agents: Transforming possibilities in technology and beyond. *ArXiv*, abs/2508.11957, 2025.

Nicholas Roth, Chris Hidey, L. Spangher, William F. Arnold, Chang Ye, Nick Masiewicki, Jinoo Baek, Peter Grabowski, and Eugene Ie. Factored agents: Decoupling in-context learning and memorization for robust tool use. *ArXiv*, abs/2503.22931, 2025.

Vinu Sankar Sadasivan, Aounon Kumar, S. Balasubramanian, Wenxiao Wang, and S. Feizi. Can ai-generated text be reliably detected? *ArXiv*, abs/2303.11156, 2023.

Haoyuan Shi, Yunxin Li, Xinyu Chen, Longyue Wang, Baotian Hu, and Min Zhang. *AniMaker: Multi-Agent Animated Storytelling with MCTS-Driven Clip Generation*. 2025.

Karanbir Singh and William Ngu. Bias-aware agent: Enhancing fairness in ai-driven knowledge retrieval. *Companion Proceedings of the ACM on Web Conference 2025*, 2025.

Guijin Son, Jiwoo Hong, Honglu Fan, Heejeong Nam, Hyunwoo Ko, Seungwon Lim, Jinyeop Song, Jinha Choi, Gonccalo Paulo, Youngjae Yu, and Stella Biderman. When ai co-scientists fail: Spot-a benchmark for automated verification of scientific research. *ArXiv*, abs/2505.11855, 2025.

Harold Triedman and Vitaly Shmatikov. Millstone: How open-minded are llms? *ArXiv*, abs/2509.11967, 2025.

Jian Wang, Boyan Zhu, Chak Tou Leong, Yongqing Li, and Wenjie Li. Scaling over scaling: Exploring test-time scaling pareto in large reasoning models. *ArXiv*, abs/2505.20522, 2025.

Jiuniu Wang, Zehua Du, Yuyuan Zhao, Bo Yuan, Kexiang Wang, Jian Liang, Yaxi Zhao, Yihen Lu, Gengliang Li, Junlong Gao, Xin Tu, and Zhenyu Guo. Aesopagent: Agent-driven evolutionary system on story-to-video production. *ArXiv*, abs/2403.07952, 2024a.

Zifeng Wang, Benjamin P. Danek, Ziwei Yang, Zheng Chen, and Jimeng Sun. Can large language models replace data scientists in biomedical research? 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

Isaac Wu and Michael Maslowski. Courtguard: A local, multiagent prompt injection classifier. *ArXiv*, abs/2510.19844, 2025.

Prasanna N. Wudali, Moshe Kravchik, Ehud Malul, P. A. Gandhi, Y. Elovici, and A. Shabtai. Rule-att&ck mapper (ram): Mapping siem rules to ttps using llms. *ArXiv*, abs/2502.02337, 2025.

Ruibin Xiong, Yimeng Chen, Dmitrii Khizbullin, Mingchen Zhuge, and Jurgen Schmidhuber. Beyond outlining: Heterogeneous recursive planning for adaptive long-form writing with language models. *ArXiv*, abs/2503.08275, 2025.

Yang Xu, Yongqiang Yao, Yufan Huang, Mengnan Qi, Maoquan Wang, Bin Gu, and Neel Sundaresan. Rethinking the instruction quality: Lift is what you need. *ArXiv*, abs/2312.11508, 2023.

Liwei Yang, Xinying Wang, Xiaotang Zhou, Zhengchao Wu, and Ningning Tan. Application of multiple chain-of-thought in contrastive reasoning for implicit sentiment analysis. *ArXiv*, abs/2503.07140, 2025a.

Yingxuan Yang, Bo Huang, Siyuan Qi, Chao Feng, Haoyi Hu, Yuxuan Zhu, Jinbo Hu, Haoran Zhao, Ziyi He, Xiao Liu, Zongyu Wang, Lin Qiu, Xuezhi Cao, Xunliang Cai, Yong Yu, and Weina Zhang. Understanding and optimizing agentic workflows via shapley value. 2025b.

Yao Yao, Z. Li, and Hai Zhao. Beyond chain-of-thought, effective graph-of-thought reasoning in language models. 2023.

Zhangyue Yin, Qiushi Sun, Zhiyuan Zeng, Zhiyuan Yu, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. Arise: An adaptive resolution-aware metric for test-time scaling evaluation in large reasoning models. *ArXiv*, abs/2510.06014, 2025.

Chaojia Yu, Zihan Cheng, Hanwen Cui, Yishuo Gao, Zexu Luo, Yijin Wang, Hangbin Zheng, and Yong Zhao. A survey on agent workflow – status and future. *2025 8th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 770–781, 2025.

Chao Zhang, Yuren Mao, Yijiang Fan, Yu Mi, Yunjun Gao, Lu Chen, Dongfang Lou, and Jinshu Lin. Finsql: Model-agnostic llms-based text-to-sql framework for financial analysis. *Companion of the 2024 International Conference on Management of Data*, 2024.

Yuanshuo Zhang, Yuchen Hou, Bohan Tang, Shuo Chen, Muhan Zhang, Xiaowen Dong, and Siheng Chen. Gnns as predictors of agentic workflow performances. *ArXiv*, abs/2503.11301, 2025.

Wei Zhao, Xin Yang, Zhihan Lyu, Cai Xu, and Ziyu Guan. Road of large language model: Source, challenge, and future perspectives. *Research*, 8, 2025.

Banghua Zhu, Jiantao Jiao, and Michael I. Jordan. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *ArXiv*, abs/2301.11270, 2023.