

Towards LLM Reliability: A System Reliability Perspective

Shaomin Wu

Kent Business School, University of Kent, Canterbury, Kent CT2 7FS UK

Abstract. The reliability of large language models (LLMs) has recently emerged as a research topic. This paper investigates it from a system reliability perspective. We hope it can shed some light for researchers in reliability engineering and LLM fields.

1 Introduction

The rapid ascent of large language models (LLMs) marks a fundamental paradigm shift in artificial intelligence (AI). Systems such as GPT-4, Claude, Gemini, and DeepSeek-R1 exhibit unprecedented capabilities in natural language understanding, code generation, and complex reasoning. These models are being rapidly deployed in high-stakes domains including medical diagnosis support, legal document analysis, financial decision-making, and autonomous vehicle planning. Yet alongside this technological progress comes an increasingly urgent question: Are they reliable?

The historical experience of reliability engineering offers an important conceptual starting point. Since the 1950 AGREE (the Advisory Group on Reliability of Electronic Equipment) report established the basic paradigm of reliability engineering (Coppola, 1984), the field has adhered to a core principle: things will eventually fail. The goal of reliability engineering is not to pursue absolute faultlessness, but to probabilistically assess and improve a system’s ability to perform specified functions under given conditions for a given time. This idea was initially applied to hardware systems such as radar and avionics (Rausand & Hoyland, 2020), then extended to software engineering, giving rise to software reliability modelling (Musa, 1979), reliability growth models (Okamura, Dohi, & Osaki, 2013), and reliability improvement through maintenance (Wu, 2019).

Bringing the perspective of reliability engineering to LLMs presents a fundamental conceptual challenge. In traditional reliability engineering, a component’s “failure” typically has a clear physical or logical criterion: open circuit, short circuit, memory error, and logical deadlock. But what is an LLM’s “failure”? When a model generates a completely fabricated reference with a fluent, confident tone, has it “failed”? When it gradually accepts a user’s erroneous premise in multi-turn dialogue and rationalises it, is it “performing its specified function” or “experiencing failure”?

The answers are not obvious. More complex still, LLMs exhibit a distinctive failure mode: their errors are often presented in highly credible form. Psychological research shows that humans tend to equate fluency of information processing with truthfulness (Reber & Schwarz, 1999). LLM-generated text is perfect in grammar, confident in tone, and coherent in logic, which precisely hijacks human cognitive shortcuts. A 2025 randomised clinical trial showed that even professional physicians, after exposure to flawed LLM advice, saw diagnostic accuracy drop to 73.3% (Qazi et al., 2026). This means that LLM reliability is not merely a technical problem but also a cognitive and trust problem.

This paper aims to systematically examine the core reliability challenges facing LLMs from the perspective of reliability engineering. We first define the conceptual scope and measurement framework for LLM reliability (Section 2), then analyse

four key failure modes: hallucination and factual failure (Section 3), calibration and uncertainty expression failure (Section 4), safety alignment and adversarial vulnerability (Section 5), and the reliability of evaluation itself (Section 6). On this basis, we discuss engineering strategies for reliability improvement and outline future research directions (Section 7).

2 Conceptual Framework for LLM Reliability

2.1 From “Correctness” to “Reliability”

In traditional software engineering, program correctness is a binary concept: a program either satisfies its specification or it does not. Reliability is a probabilistic concept: the probability that a system operates without failure under given conditions for a given time interval. For LLMs, the concept of “correctness” becomes ambiguous: the same question may have multiple “correct” answers; model outputs are the result of stochastic sampling; task “specifications” are often given in natural language and are themselves ambiguous.

Therefore, LLM reliability must shift from a “correctness” framework to a “reliability” framework. This means we no longer ask “Is this model correct?” but rather “In what proportion of situations, with what degree of uncertainty, does this model produce acceptable results?” This requires probabilistic measures, tolerance concepts, and criteria for defining “acceptable failure” for specific application domains.

2.2 Multidimensional Structure of Reliability

Drawing on established reliability-engineering concepts concerning reliability, durability, maintainability, and reliability-oriented design (Dhillon, 1999), we structure the reliability of LLMs along four complementary dimensions as follows:

- **Factual Reliability:** The probability that information generated by an LLM satisfies the factual requirements of its intended application under specified conditions. In practice, this dimension can be operationalised through measures of factual precision, claim-level verification, and hallucination rate (Huang et al., 2025; Min et al., 2023; Wang et al., 2024).
- **Calibration Reliability:** The degree to which the confidence expressed by a model corresponds to the empirical probability that its output is correct. A well-calibrated model assigning approximately 90% confidence to a set of predictions should be correct for approximately 90% of those predictions (Jiang, Araki, Ding, & Neubig, 2021; Lin, Hilton, & Evans, 2022; Tian et al., 2023).
- **Safety Reliability:** The model’s ability to appropriately refuse harmful or disallowed requests and maintain safe behaviour under adversarial inputs, including attempts to circumvent its safety safeguards through jailbreaks (Bai et al., 2022; Z. Dong, Zhou, Yang, Shao, & Qiao, 2024; Xu, Liu, Deng, Li, & Picek, 2024; Yi et al., 2024).
- **Evaluation Reliability:** An important but often overlooked issue is whether the benchmarks, metrics, and evaluation procedures used to assess model reliability are themselves reliable and valid measures of the properties they are intended to capture (Choi, Park, Cho, Park, & Kim, 2026; Lior, Habba, Levy, Caciularu, & Stanovsky, 2025; Lunardi, Mea, Mizzaro, & Roitero, 2025; Xiao, Zhang, Lai, & Liao, 2023)

These four dimensions are not independent but mutually intertwined. For example, a model may be highly accurate at the factual level but poorly calibrated (overconfident in a minority of errors), making its overall reliability lower than surface accuracy suggests.

2.3 Performance-Reliability Tension

In LLM development, there exists a tension analogous to the “performance-interpretability tradeoff” in the XAI field (Rudin, 2019). Larger parameter scales, more complex training objectives, and more aggressive inference strategies (such as chain-of-thought, self-reflection) generally improve task performance but may come at the cost of calibration quality and safety reliability. Understanding and managing this tension is a core task of LLM reliability engineering.

3 Hallucination: Systematic Analysis of Factual Failure

3.1 Definition and Classification of Hallucination

Hallucination is the most widely known and destructive failure mode of LLMs. Its core characteristic is: the model generates fluent, confident content that is inconsistent with facts (Huang et al., 2025; Tonmoy et al., 2024; Zhang et al., 2025). From a reliability engineering perspective, hallucination is a “soft failure”: the system is still running, still producing output, but the output does not meet specified functional requirements.

Recent research has proposed multi-level hallucination taxonomies. By source, they can be classified as: hallucinations caused by training data bias, hallucinations caused by knowledge boundary ambiguity, and hallucinations caused by context misunderstanding. By form, they can be classified as: input-conflicting hallucination (deviating from user input), context-conflicting hallucination (inconsistent with the model’s own prior generation), and fact-conflicting hallucination (contradicting established world knowledge) (Zhang et al., 2025).

A recent paper argues that Transformer-based architectures are structurally incapable of verification (V), and that the absence of consequence-bearing experience (E) renders LLMs fundamentally untrustworthy regardless of scale or benchmark performance (Devin, 2026). This “KEV constraint” (Knowledge is massive, Experience is absent, Verification is absent) posits that scaling intensifies rather than resolves the failure mode, which the authors term “Artificially Confabulating and Untrustworthy” (ACU) (Devin, 2026). Devin (2026) also documents a compounding failure stack: Speed-Weighted Reasoning Collapse → Architectural Confabulation → Uncalibrated Confidence.

3.2 Quantitative Evaluation of Hallucination

Evaluating hallucination requires an operationalizable measurement system. Traditional NLP metrics (BLEU, ROUGE, BERTScore) measure text similarity, not factual correctness (where NLP stands for Natural Language Processing). A response with completely different wording from the reference answer may be factually correct, while a response with high overlap may contain critical factual errors.

The development of factuality-specific metrics represents important recent progress. FactScore measures the factual precision of long-form text generated by large language models (Min et al., 2023); FEVER Score evaluates accuracy of fact verification (Thorne, Vlachos, Christodoulopoulos, & Mittal, 2018); NLI-based methods use a Natural Language Inference (NLI) (or entailment) model to check whether a generated statement is logically supported by a given source text (W.-F. Chen, Zhao, Karimi, & Flek, 2025; Sathe & Park, 2021). Recent research has further proposed entity-centric evaluation methods that, in scenarios such as medical question answering, extract key medical entities from generated responses to measure their coverage of input context and responsiveness to diagnostic focus (Liu & Kolachalama, 2025).

A comprehensive survey of hallucination detection methods categorises approaches into several types: those based on external databases, those leveraging the model’s internal knowledge, and those based on consistency across multiple

outputs (Huang et al., 2025). The paper also presents a comprehensive overview of hallucination detection benchmarks, including TruthfulQA, HaluEval, HalluQA, and others, with coverage of different languages, hallucination types, and evaluation metrics (Huang et al., 2025).

3.3 Mitigation Strategies and Their Limitations

Retrieval-Augmented Generation (RAG) has emerged as one of the most widely adopted approaches for mitigating hallucinations in LLMs by grounding generation in externally retrieved information (Li, Fu, Verma, Buitelaar, & Liu, 2025). However, RAG does not eliminate hallucination; rather, it introduces additional failure modes associated with the retrieval and utilisation of external evidence. These include retrieval failures arising from imperfect retrieval quality or query–document mismatch, as well as generation failures in which the model ignores, misinterprets, or insufficiently utilises retrieved information, even when the retrieved context is relevant and factually correct (Huang et al., 2025; Ni et al., 2026; Shi et al., 2026; Yeh, Li, & Mallick, 2026).

Post-hoc verification and correction methods constitute another path. These methods perform secondary checks on model outputs, identify unreliable claims, and correct or flag them (Pan et al., 2023). Their advantage is model-agnosticism, applicable to any black-box LLM. However, post-hoc methods face a fundamental limitation: verification itself requires reliability (Kamoi, Zhang, Zhang, Han, & Zhang, 2024; Norman, Rivera, & Hughes, 2026). If the verifier shares similar biases or knowledge blind spots with the generator, verification may fail. A recent study introduces the concept of “generator–evaluator self-consistency” and finds a “consistency dilemma”: models with higher self-consistency may actually be more vulnerable to systematic mistakes (Mancoridis & Hitzig, 2026).

From a reliability engineering perspective, we should recognise: there is no “zero-hallucination” LLM. Just as there is no “zero-failure” engineering system. The key question is not to eliminate hallucination but to bring its occurrence rate, severity, and detectability to acceptable levels and to establish corresponding failure detection and mitigation mechanisms.

4 Calibration and Uncertainty Expression: The Neglected Reliability Dimension

4.1 The Nature of the Calibration Problem

Calibration is one of the core dimensions of reliability. A well-calibrated forecasting system should have its forecast probabilities consistent with actual outcome frequencies: events predicted with “81% probability” should occur approximately 81% of the time. For LLMs, the calibration problem manifests as systematic misalignment between confidence and accuracy (Jiang et al., 2021).

LLM calibration problems have a unique form. Traditional classification model calibration typically targets the confidence of a single prediction; LLM “confidence” must be inferred from generated text—the model’s tone, the degree of certainty in wording, whether it expresses uncertainty. Research shows that LLMs exhibit significant systematic bias in expressing uncertainty: they tend to be overconfident, especially when facing questions beyond their training distribution (Xiong et al., 2024).

A study on chain-of-thought confidence estimation proposes the Enhanced Dirichlet and Topology Risk (EDTR) method, which treats each chain-of-thought as a vector in high-dimensional space and extracts topological risk features capturing the geometric structure of reasoning distributions. The method achieves 41% better calibration than competing methods (More et al., 2025). Another study investigates the confidence-faithfulness gap using steering vectors to manipulate

verbalised confidence in LLMs, demonstrating that accuracy and verbalised confidence are linearly encoded but nearly orthogonal in representation space (Miao & Ungar, 2026).

4.2 Empirical Evidence of Calibration Failure

A recent large-scale evaluation revealed a severe calibration problem in predictions of indeterminate future events. In particular, Gemini’s stated 90% confidence intervals captured actual outcomes in only 29.7% of cases, while its average interval for the S&P 500 was approximately 98 points wide, compared with an expected width of approximately 500 points (Pandit, Pandit, et al., 2026).

More concerning is that calibration failure also exists in contexts where models claim to “know what they don’t know.” When asked about events after their training cutoff, models may respond with a seemingly cautious but actually overconfident tone rather than clearly acknowledging knowledge boundaries.

4.3 Uncertainty Quantification Methods

Uncertainty quantification (UQ) methods for LLMs can be categorised into several types: self-report, consistency-based, logit-based, entropy-based, and ensemble-based. For example, self-report-based methods have the model directly express its confidence in its answer; consistency-based methods compare consistency across multiple generated results; and logit-based methods use the model’s output probability distribution to compute uncertainty measures (Moore, Roberts, Watson, & Wisniewski, 2025).

Each method has its applicable scenarios and limitations. For example, self-report is constrained by the model’s metacognitive ability regarding its own knowledge boundaries; and consistency methods may be misled by systematic biases. Within the reliability engineering framework, the value of uncertainty quantification lies not in itself but in whether it can drive effective reliability decisions: triggering human intervention, retrieving external knowledge, or refusing to answer when uncertainty exceeds a threshold.

5 Safety Reliability: Behavioural Boundaries Under Adversarial Conditions

5.1 Threat Model

LLM safety research has deep connections to “adversarial failure” analysis in reliability engineering. When models are deployed in open environments, they face not only random failures but also deliberate manipulation: users or third parties attempting to induce harmful outputs.

Jailbreaking refers to bypassing model safety alignment through carefully designed prompts to induce prohibited content. Prompt injection involves hijacking model behaviour by tampering with task instructions or context (e.g., malicious web pages, document content). A comprehensive survey of jailbreak research from 2023–2025 groups techniques into five main categories: prompt-based injections, role-play conditioning, multi-turn dialogue, multilingual or multimodal exploits, and optimization-driven pipelines (Knowlton, Campa, Gallo, Dajani, & Alzahrani, 2026). Discovery methods include human red teaming and automated attacker systems such as AutoJailbreak (Lu et al., 2024) and LLMStinger (Jha, Arora, & Ganesh, 2026).

These attack forms are closely related to the instruction-following nature of LLMs: models are trained to “respond helpfully to instructions,” but this objective exists in tension with safety constraints.

5.2 Metrics for Attack and Defence

Safety reliability assessment requires explicit threat models and measurement systems. Attack success rate (ASR) measures the proportion of effective attacks; transferability measures the effectiveness of attacks designed for one model on another; performance degradation rate (PDR) measures task performance degradation caused by adversarial inputs.

A comprehensive survey classifies LLM jailbreak defences into two broad categories: prompt-level defences, which filter or preprocess potentially malicious inputs, and model-level defences, which modify or fine-tune the model to strengthen its intrinsic safety alignment (Yi et al., 2024). From a reliability engineering perspective, no single defence is complete. Effective safety reliability requires defence-in-depth and explicit criteria for residual risk acceptance.

6 The Reliability Problem of Evaluation Reliability

6.1 Fragility of Benchmarks

The benchmarks used to evaluate LLM reliability may themselves be unreliable. Recent studies have identified pervasive label errors and ambiguities in widely used LLM benchmarks, which can obscure model failures and compromise reliability assessments (Vendrow, Vendrow, Beery, & Madry, 2025). Benchmark contamination provides another threat: evaluation data may overlap with model training data, potentially inflating reported performance (Balloccu, Schmidová, Lango, & Dusek, 2024; Deng, Zhao, Tang, Gerstein, & Cohan, 2024). Moreover, benchmark performance may not generalise to deployment settings because the distribution of real-world user prompts can differ substantially from that represented in evaluation benchmarks (Seegmiller & Preum, 2026).

Recent evaluations have exposed serious weaknesses in the integrity of LLM and agent benchmarks. Automated red-teaming has demonstrated that agents can exploit benchmark vulnerabilities to achieve near-perfect scores without solving the underlying tasks, with BenchJack identifying 219 distinct flaws across ten agent benchmarks (H. Wang et al., 2026). Independent evaluation work has likewise documented substantial reward-hacking behaviour in frontier models, including attempts to manipulate evaluation harnesses (METR, 2026). Benchmark defects have also led to the withdrawal or retraction of recommendations for widely used coding benchmarks: OpenAI discontinued reporting on SWE-bench Verified after finding substantial contamination and test-design problems, and subsequently retracted its recommendation to adopt SWE-Bench Pro after finding that approximately 30% of its tasks were broken (OpenAI, 2026a, 2026b).

Balepur et al. (2026) introduce BenchMarker, an education-inspired toolkit that evaluates multiple-choice benchmark items along three dimensions: contamination (whether items appear online), shortcuts (whether shallow cues in the answer choices enable guessing), and writing errors (violations of a 19-rule educational rubric). Auditing 12 benchmarks, they find substantial flaws, including widespread online occurrence of TruthfulQA items, shortcut cues in SocialQA, and pervasive writing-rule violations in HellaSwag.

6.2 The Verifier Is Part of the System

Research has shown that strong LLM judges can achieve over 80% agreement with human preferences, while also exhibiting systematic biases including position bias, verbosity bias, and self-enhancement bias (Zheng et al., 2023). These biases are tolerable noise when a judge ranks a static set of outputs but become consequential when the judge is used to certify reliability.

Recent research suggests that LLM-as-judge evaluations are themselves sensitive to both the evaluator model and the design of the evaluation prompt. In particular, different prompting strategies—including the use of self-generated evaluation steps, evaluation metrics, and prompt-component ordering—can substantially affect the resulting evaluation outcomes (Wen et al., 2025). Thus, LLM-based evaluation does not provide an independent guarantee of reliability merely because the evaluation process is automated.

6.3 Toward a Reliable Evaluation Framework

A recent study proposes the metric *reliable@k* to quantify whether models exhibit consistent competence across “cousin prompts” that convey analogous user intents but with subtle nuances. Across 20 proprietary and 26 open-source LLMs, the study finds that current models exhibit substantial insufficiency in nuance-oriented reliability, with performance dropping by up to 61.8% following nuanced prompt modifications (J. Dong et al., 2026).

From a reliability engineering perspective, a single accuracy or pass rate metric is insufficient. We need:

- **Distribution information:** not just average performance, but also variance and tail risk (worst-case performance).
- **Calibration information:** whether model confidence is reliable.
- **Robustness information:** the degree of performance degradation under input perturbation, distribution shift, and adversarial conditions.
- **Failure mode analysis:** when models fail, how severe and detectable is the failure?

Building reliable LLM evaluation frameworks requires methodological self-awareness. This includes: using multiple independent benchmarks to reduce dependence on a single evaluation tool; adopting hybrid human-automated evaluation methods; conducting sensitivity analysis of evaluation results (sensitivity of conclusions to evaluator choice, prompt design, scoring criteria); and incorporating threat models into evaluation design—explicitly specifying the failure types and adversarial conditions targeted by evaluation.

7 Strategies and Prospects for Reliability Engineering

7.1 From Component Reliability to System Reliability

LLMs are typically deployed as one component within a larger system. The basic principles, including redundancy, isolation, fault detection and recovery, graceful degradation, of system reliability engineering are equally applicable to LLM systems.

A recent paper develops an in-depth system-level taxonomy of fifteen hidden failure modes in practical LLM applications, such as multi-step reasoning drift, latent inconsistency, context-boundary degradation, incorrect tool invocation, version drift, and cost-driven performance collapse (Vinay, 2026). The paper argues that benchmarks are designed to evaluate knowledge or reasoning but can say very little about stability, reproducibility, drift, or ease of workflow integration.

Recent work suggests that role-separated multi-agent architectures may reduce correlated errors by assigning distinct reasoning and evaluation roles to different agents, thereby improving system-level reliability (Pandit et al., 2026). If multiple LLM instances reason about the same problem in the same way, their errors may be correlated (similar training

data, similar architecture leading to similar blind spots). By assigning different instances, different reasoning roles or perspectives, error correlation can be reduced, improving system-level reliability (Kim, Garg, Peng, & Garg, 2025).

Human-in-the-loop design acknowledges LLM imperfection and introduces human judgment at critical decision points. Automation bias can undermine human oversight when users defer to erroneous LLM recommendations, even when they have received AI training. Recent work therefore emphasises interface mechanisms that promote critical engagement, including uncertainty cues, explanatory information, and cognitive-forcing mechanisms designed to introduce epistemic friction (Z. Chen, Luo, & Sra, 2025; Jovchevski, Buijsman, & Neerincx, 2026; Qazi et al., 2025).

A recent framework called LLMOps presents a continuous evaluation framework specifically for LLM-based systems, providing a multi-layered protocol for differentiating traditional data drift from emergent prompt/semantic drift. Empirically, the framework achieves F1-score improvement from 0.89 to 0.95, hallucination rate reduction from 12% to 5%, and 94% reduction in incident management decision latency enabled by automated rollback and LLM-as-a-judge mechanisms (Kommuri, 2026).

7.2 Prospects for Reliability Growth

The concept of “reliability growth testing” in reliability engineering, which continuously improving system reliability through test-analyse-fix cycles or in-field maintenance, can this be transferred to LLMs? In traditional hardware, failure analysis leads to physical design changes. In LLMs, “fixes” may mean fine-tuning, retrieval augmentation, inference-time intervention, or system-level compensation mechanisms.

This analogy has its limitations: LLM capability boundaries are fuzzy, making “failure” difficult to define; fixes may introduce new failure modes (catastrophic forgetting, alignment tax). But the core idea, which drives continuous improvement through systematic failure analysis, retains value.

7.3 Open Challenges

There are some open challenges, as illustrated below.

- **Reliability Under Distribution Shift:** Input distributions encountered by LLMs in deployment may differ significantly from training distributions. How can we detect distribution shift and quantify its impact on reliability?
- **Reliability in Multimodal and Embodied Scenarios:** As LLMs combine with vision and action systems, reliability analysis must extend to cross-modal failures and physical-world consequences.
- **Theoretical Frameworks for Reliability Guarantees:** Do theoretical tools analogous to confidence bounds in hardware reliability or software reliability growth models exist that can provide formal guarantees for LLM reliability?
- **Human Factors Dimensions of Reliability:** How can we design human-AI interaction so that humans can appropriately calibrate their trust in LLMs, neither over-relying nor over-suspecting?
- **Researchers on LLM Reliability:** Traditional system reliability research can proceed without deep knowledge of a system’s internal operation or its repair processes; reliability is assessed largely through observable failure behaviour over time. LLM reliability, however, demands a substantially wider knowledge base: spanning machine learning, linguistics, and human cognition, and more sophisticated experimental methodologies to verify results, because the sources of failure are opaque, context-dependent, and often not reproducible in the conventional sense.

8 Conclusion

The reliability of large language models is a threshold that artificial intelligence must cross as it moves from the laboratory to critical social infrastructure. Examining this problem from the perspective of reliability engineering gives us not merely borrowed terminology but a systematic way of thinking: acknowledging the inevitability of failure, quantifying the probability and consequences of failure, and designing systems that can detect, mitigate, and recover from failure.

Hallucination, calibration failure, adversarial vulnerability, and evaluation unreliability constitute the core challenges of current LLM reliability. The resolution of these challenges will not come from a single technological breakthrough but requires the coordinated evolution of evaluation methods, system architectures, human-computer interaction, and governance frameworks.

The reliability engineering of large language models has only just begun. As the reliability pioneers of the 1950s recognised when facing complex electronic systems, our goal is not to build machines that never fail but to understand, measure, and manage failure—to improve, in a probabilistic sense, a system’s ability to perform specified functions under given conditions. For LLMs, this goal is both urgent and intellectually challenging.

References

- Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., . . . Kaplan, J. (2022). Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*. doi: <https://doi.org/10.48550/arXiv.2212.08073>
- Balepur, N., Rajasekaran, B., Oh, H. J., Xie, M., Desai, A., Gupta, V., . . . Boyd-Graber, J. L. (2026, July). BenchMarker: An education-inspired toolkit for highlighting flaws in multiple-choice benchmarks. In *Proceedings of the 64th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 15793–15824). San Diego, California, United States: Association for Computational Linguistics. doi: <https://doi.org/10.18653/v1/2026.acl-long.719>
- Balloccu, S., Schmidová, P., Lango, M., & Dusek, O. (2024, March). Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In Y. Graham & M. Purver (Eds.), *Proceedings of the 18th conference of the european chapter of the association for computational linguistics (volume 1: Long papers)* (pp. 67–93). St. Julian’s, Malta: Association for Computational Linguistics. doi: <https://doi.org/10.18653/v1/2024.eacl-long.5>
- Chen, W.-F., Zhao, Z., Karimi, A., & Flek, L. (2025). Explainable hallucination through natural language inference mapping. In *Findings of the association for computational linguistics: Acl 2025* (pp. 1888–1896). doi: <https://doi.org/10.18653/v1/2025.findings-acl.96>
- Chen, Z., Luo, Y., & Sra, M. (2025). Engaging with ai: How interface design shapes human-ai collaboration in high-stakes decision-making. *arXiv preprint arXiv:2501.16627*. doi: <https://doi.org/10.48550/arXiv.2501.16627>
- Choi, J., Park, S., Cho, C., Park, H., & Kim, B. (2026). Diagnosing the reliability of LLM-as-a-judge via item response theory. *arXiv preprint arXiv:2602.00521*. doi: <https://doi.org/10.48550/arXiv.2602.00521>
- Coppola, A. (1984). Reliability engineering of electronic equipment: a historical perspective. *IEEE Transactions on Reliability*, 33(1), 29–35. doi: <https://doi.org/10.1109/TR.1984.6448271>
- Deng, C., Zhao, Y., Tang, X., Gerstein, M., & Cohan, A. (2024). Investigating data contamination in modern benchmarks for large language models. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Proceedings of the 2024 conference*

- of the north american chapter of the association for computational linguistics: *Human language technologies (volume 1: Long papers)* (pp. 8706–8719). Mexico City, Mexico: Association for Computational Linguistics. doi: <https://doi.org/10.18653/v1/2024.naacl-long.482>
- Devin, A. J. (2026). Artificial confabulation and untrustworthiness: Verification, reliance and the structural limits of generative AI. *KEV Academic Series, version, 3*, 17. doi: <https://doi.org/10.5281/zenodo.21409378>
- Dhillon, B. S. (1999). *Design reliability: fundamentals and applications*. CRC press. doi: <https://doi.org/10.1201/9780367802400>
- Dong, J., Zhang, Y., Yan, L., Zhong, Z., Wei, T., Zhang, C., & Qiu, H. (2026). Revisiting the reliability of language models in instruction-following. In *Proceedings of the 64th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 7784–7812). doi: <https://doi.org/10.18653/v1/2026.acl-long.354>
- Dong, Z., Zhou, Z., Yang, C., Shao, J., & Qiao, Y. (2024). Attacks, defenses and evaluations for LLM conversation safety: A survey. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies*. doi: <https://doi.org/10.18653/v1/2024.naacl-long.375>
- Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., . . . Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2), 1–55. doi: <https://doi.org/10.1145/3703155>
- Jha, P., Arora, A., & Ganesh, V. (2026). *Llmstinger: Jailbreaking llms using rl fine-tuned llms*. doi: <https://doi.org/10.48550/arXiv.2411.08862>
- Jiang, Z., Araki, J., Ding, H., & Neubig, G. (2021). How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9, 962–977. doi: https://doi.org/10.1162/tac1_a.00407
- Jovchevski, P., Buijsman, S., & Neerincx, M. (2026). What is wrong with automation bias? *Philosophy & Technology*, 39(2), 84. doi: <https://doi.org/10.1007/s13347-026-01090-9>
- Kamoi, R., Zhang, Y., Zhang, N., Han, J., & Zhang, R. (2024). When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12, 1417–1440. doi: https://doi.org/10.1162/tac1_a.00713
- Kim, E., Garg, A., Peng, K., & Garg, N. (2025). Correlated errors in large language models. *arXiv preprint arXiv:2506.07962*. doi: <https://doi.org/10.48550/arXiv.2506.07962>
- Knowlton, B., Campa, J., Gallo, D. S., Dajani, K., & Alzahrani, N. (2026). Prompt-based jailbreaking of leading LLM chatbots: A survey of attacks and defenses. *IEEE Transactions on Artificial Intelligence*, 7(8), 4237–4251. doi: <https://doi.org/10.1109/TAI.2026.3665656>
- Kommuri, S. S. (2026). Llmops: A continuous evaluation framework with automated rollback and a/b testing for production genai systems. In *2026 technology and innovation conference (tic)* (p. 1-5). doi: <https://doi.org/10.1109/TIC68483.2026.11702875>
- Li, Y., Fu, X., Verma, G., Buitelaar, P., & Liu, M. (2025). Mitigating hallucination in large language models (llms): an application-oriented survey on rag, reasoning, and agentic systems. *arXiv preprint arXiv:2510.24476*. doi: <https://doi.org/10.48550/arXiv.2510.24476>
- Lin, S., Hilton, J., & Evans, O. (2022). Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*. doi: <https://doi.org/10.48550/arXiv.2205.14334>
- Lior, G., Habba, E., Levy, S., Caciularu, A., & Stanovsky, G. (2025). Reliableeval: A recipe for stochastic LLM

- evaluation via method of moments. In *Emnlp (findings)* (pp. 11146–11153). doi: <https://doi.org/10.18653/v1/2025.findings-emnlp.594>
- Liu, Y., & Kolachalama, V. B. (2025). Entity-centric evaluation of large language model responses for medical question-answering tasks. *medRxiv*. doi: <https://doi.org/10.1101/2025.11.12.25340106>
- Lu, L., Yan, H., Yuan, Z., Shi, J., Wei, W., Chen, P.-Y., & Zhou, P. (2024). *Autojailbreak: Exploring jailbreak attacks and defenses through a dependency lens*. doi: <https://doi.org/10.48550/arXiv.2406.03805>
- Lunardi, R., Mea, V. D., Mizzaro, S., & Roitero, K. (2025). On robustness and reliability of benchmark-based evaluation of LLMs. In *Ecai 2025: 28th european conference on artificial intelligence 25-30 october 2025, bologna, italy including 14th conference on prestigious applications of intelligent systems (pais 2025) proceedings* (pp. 4603–4610). doi: <https://doi.org/10.3233/FAIA251363>
- Mancoridis, M., & Hitzig, Z. (2026). The consistency dilemma in llms: Generator-evaluator agreement and vulnerability to mistakes. *arXiv preprint arXiv:2606.30653*. doi: <https://doi.org/10.48550/arXiv.2606.30653>
- METR. (2026, 05). *Frontier risk report (february to march 2026)*. <https://metr.org/blog/2026-05-19-frontier-risk-report/>.
- Miao, M. M., & Ungar, L. (2026). Closing the confidence-faithfulness gap in large language models. *arXiv preprint arXiv:2603.25052*. doi: <https://doi.org/10.48550/arXiv.2603.25052>
- Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.-t., Koh, P. W., . . . Hajishirzi, H. (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* (pp. 12076–12100). doi: <https://doi.org/10.18653/v1/2023.emnlp-main.741>
- Moore, K., Roberts, J., Watson, D., & Wisniewski, P. (2025). Investigating human-aligned large language model uncertainty. *arXiv preprint arXiv:2503.12528*. doi: <https://doi.org/10.48550/arXiv.2503.12528>
- More, A., Zhang, A., Bonilla, N., Vivekan, A., Zhu, K., Sharafolleslami, P., & Chaudhary, M. (2025). Optimizing chain-of-thought confidence via topological and dirichlet risk analysis. *arXiv preprint arXiv:2511.06437*. doi: <https://doi.org/10.48550/arXiv.2511.06437>
- Musa, J. D. (1979). Software reliability measurement. *Journal of Systems and Software*, 1, 223–241. doi: [https://doi.org/10.1016/0164-1212\(79\)90023-2](https://doi.org/10.1016/0164-1212(79)90023-2)
- Ni, B., Liu, Z., Lei, Y., Wang, L., Zhao, Y., Cheng, X., . . . others (2026). Towards trustworthy retrieval augmented generation for large language models: A survey. *ACM Computing Surveys*, 58(15), 1–36. doi: <https://doi.org/10.1145/3837074>
- Norman, J. D., Rivera, M. U., & Hughes, D. A. (2026). Reliability without validity: A systematic, large-scale evaluation of llm-as-a-judge models across agreement, consistency, and bias. *arXiv preprint arXiv:2606.19544*. doi: <https://doi.org/10.48550/arXiv.2606.19544>
- Okamura, H., Dohi, T., & Osaki, S. (2013). Software reliability growth models with normal failure time distributions. *Reliability Engineering & System Safety*, 116, 135–141. doi: <https://doi.org/10.1016/j.ress.2012.02.002>
- OpenAI. (2026a, Jul). *Separating signal from noise in coding evaluations*. Retrieved from <https://openai.com/index/separating-signal-from-noise-coding-evaluations/>
- OpenAI. (2026b, Feb). *Why SWE-bench verified no longer measures frontier coding capabilities*. Retrieved from <https://openai.com/index/why-we-no-longer-evaluate-swe-bench-verified/>
- Pan, L., Saxon, M., Xu, W., Nathani, D., Wang, X., & Wang, W. Y. (2023). Automatically correcting large language

- models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*. doi: <https://doi.org/10.48550/arXiv.2308.03188>
- Pandit, V., Pandit, A., et al. (2026). *The reliability chasm: AI accuracy across three reasoning domains*. doi: <https://doi.org/10.21203/rs.3.rs-9070952/v1>
- Qazi, I. A., Ali, A., Khawaja, A. U., Akhtar, M. J., Sheikh, A. Z., & Alizai, M. H. (2025). Automation bias in large language model assisted diagnostic reasoning among AI-trained physicians. *medRxiv*, 2025–08. doi: <https://doi.org/10.1101/2025.08.23.25334280>
- Qazi, I. A., Ali, A., Khawaja, A. U., Akhtar, M. J., Sheikh, A. Z., & Alizai, M. H. (2026). Automation bias in large language model–assisted diagnostic reasoning among physicians trained in AI literacy—a randomized clinical trial. *Nejm Ai*, 3(5), AIoa2501001. doi: <https://doi.org/10.1056/AIoa2501001>
- Rausand, M., & Hoyland, A. (2020). *System reliability theory: models, statistical methods, and applications* (Vol. 396). John Wiley & Sons. doi: <https://doi.org/10.1002/9781119373940>
- Reber, R., & Schwarz, N. (1999). Effects of perceptual fluency on judgments of truth. *Consciousness and Cognition*, 8(3), 338–342. doi: <https://doi.org/10.1006/ccog.1999.0386>
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215. doi: <https://doi.org/10.1038/s42256-019-0048-x>
- Sathe, A., & Park, J. (2021). Automatic fact-checking with document-level annotations using bert and multiple instance learning. In *Proceedings of the fourth workshop on fact extraction and verification (fever)* (pp. 101–107). doi: <https://doi.org/10.18653/v1/2021.fever-1.11>
- Seegmiller, P., & Preum, S. M. (2026, July). Measuring distribution shift in user prompts and its effects on LLM performance. In M. Liakata, V. P. Moreira, J. Zhang, & D. Jurgens (Eds.), *Proceedings of the 64th annual meeting of the Association for Computational Linguistics (volume 1: Long papers)* (pp. 32685–32704). San Diego, California, United States: Association for Computational Linguistics. doi: <https://doi.org/10.18653/v1/2026.acl-long.1508>
- Shi, J., Xie, Q., Hou, Q., Wang, Z., Su, W., Zhao, J., . . . He, T. (2026). Coda: Restoring contextual dominance via copy-encouraged attention intervention for mitigating rag hallucinations. In *Findings of the association for computational linguistics: Acl 2026* (pp. 11879–11892). doi: <https://doi.org/10.18653/v1/2026.findings-acl.576>
- Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (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)* (pp. 809–819). doi: <https://doi.org/10.18653/v1/N18-1074>
- Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., . . . Manning, C. (2023). Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 5433–5442). Association for Computational Linguistics. doi: <https://doi.org/10.18653/v1/2023.emnlp-main.330>
- Tonmoy, S. M. T. I., Zaman, S. M. M., Jain, V., Rani, A., Rawte, V., Chadha, A., & Das, A. (2024). A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*. doi: <https://doi.org/10.48550/arXiv.2401.01313>
- Vendrow, J., Vendrow, E., Beery, S., & Madry, A. (2025). Do large language model benchmarks test reliability? *arXiv preprint arXiv:2502.03461*. doi: <https://doi.org/10.48550/arXiv.2502.03461>
- Vinay, V. (2026, May). A system-level taxonomy of failure modes in large language model applications. In *2026 ieee*

- conference on artificial intelligence (CAI)* (p. 715-722). Los Alamitos, CA, USA: IEEE Computer Society. doi: <https://doi.ieeecomputersociety.org/10.1109/CAI68641.2026.11536235>
- Wang, Wang, M., Manzoor, M. A., Liu, F., Das, G. N. G. R. J., & Nakov, P. (2024). Factuality of large language models: A survey. In *Proceedings of the 2024 conference on empirical methods in natural language processing* (pp. 19519–19529). doi: <https://doi.org/10.18653/v1/2024.emnlp-main.1088>
- Wang, H., Li, H., Mang, Q., Cheung, A., Sen, K., & Song, D. (2026). *Do androids dream of breaking the game? systematically auditing ai agent benchmarks with benchjack*. doi: <https://doi.org/10.48550/arXiv.2605.12673>
- Wen, B., Ke, P., Sun, Y., Wang, C., Gu, X., Zhou, J., . . . Huang, M. (2025). HPSS: Heuristic prompting strategy search for llm evaluators. In *Findings of the association for computational linguistics: Acl 2025* (pp. 24974–25007). doi: <https://doi.org/10.18653/v1/2025.findings-acl.1282>
- Wu, S. (2019). A failure process model with the exponential smoothing of intensity functions. *European Journal of Operational Research*, 275(2), 502–513.
- Xiao, Z., Zhang, S., Lai, V., & Liao, Q. V. (2023). Evaluating evaluation metrics: A framework for analyzing nlg evaluation metrics using measurement theory. In *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 10967–10982). doi: <https://doi.org/10.18653/v1/2023.emnlp-main.676>
- Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., & Hooi, B. (2024). *Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs*. doi: <https://doi.org/10.48550/arXiv.2306.13063>
- Xu, Z., Liu, Y., Deng, G., Li, Y., & Picek, S. (2024). A comprehensive study of jailbreak attack versus defense for large language models. In *Findings of the association for computational linguistics: Acl 2024* (pp. 7432–7449). Association for Computational Linguistics. doi: <https://doi.org/10.18653/v1/2024.findings-acl.443>
- Yeh, M.-H., Li, S., & Mallick, T. (2026). Lumina: Detecting hallucinations in rag system with context–knowledge signals. In *International conference on learning representations* (Vol. 2026, pp. 151430–151448). doi: <https://doi.org/10.48550/arXiv.2509.21875>
- Yi, S., Liu, Y., Sun, Z., Cong, T., He, X., Song, J., . . . Li, Q. (2024). *Jailbreak attacks and defenses against large language models: A survey*. doi: <https://doi.org/10.48550/arXiv.2407.04295>
- Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., . . . others (2025). Siren’s song in the AI ocean: A survey on hallucination in large language models. *Computational Linguistics*, 51(4), 1373–1418. doi: <https://doi.org/10.1162/coli.a.16>
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., . . . others (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36, 46595–46623. doi: <https://doi.org/10.52202/075280-2020>