International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

The Metrology Imperative: The Necessity of Robust Evaluation Frameworks and Comprehensive Automated Judges in Generative AI

Ankur Partap Kotwal

Submitted:24/02/2026 **Revised**: 02/04/2026 **Accepted:** 10/04/2026

Abstract: Across the past several years, the accelerating advancement of Large Language Models (LLMs) and generative artificial intelligence has quietly produced a crisis that much of the field has been slow to name directly—a breakdown in the ability to evaluate what these systems can and cannot actually do. Traditional, static benchmarking methodologies have proven structurally inadequate, collapsing under the combined weight of rapid benchmark saturation, pervasive data contamination, and the kind of systematic overfitting that emerges whenever commercial incentives are tied too tightly to leaderboard rankings. This brief argues, with considerable urgency, that building robust and dynamic evaluation frameworks alongside sophisticated automated judges—most prominently through the LLM-as-a-Judge paradigm—is not an optional enhancement to existing practices but an absolute prerequisite for the continued, safe, and value-aligned development of AI systems. Through a careful examination of where current evaluation practices fail, an analysis of the architectural requirements governing automated multi-agent juries, and a survey of multi-dimensional safety assessment approaches, a coherent pathway toward genuinely reliable AI metrology is charted here. The arguments and architectural outlines presented across these sections are intended to serve as a structured foundational blueprint for a full-length 40-page journal article that will pursue the theoretical, empirical, and architectural dimensions of this problem in considerably greater depth.

Keywords: *Large Language Models; AI Evaluation; Benchmark Saturation; Data Contamination; LLM-as-a-Judge; Multi-Agent Jury; Reward Models; RLHF; Dynamic Benchmarking; Automated Red Teaming; Ethical Auditing; AI Safety; Generative AI; Goodhart's Law and AI Metrology*

List of Abbreviations: LLM: Large Language Model; NLP: Natural Language Processing; RLHF: Reinforcement Learning from Human Feedback; AI: Artificial Intelligence; CoT: Chain-of-Thought.

1. Introduction: The AI Evaluation Crisis

The quality of the instruments used to measure artificial intelligence has always gated its progress in some fundamental sense. For much of the field's history, that measurement problem was addressed through static datasets—GLUE, SQuAD, and ImageNet—whose fixed contents allowed researchers to track model performance over time in a relatively controlled way. The arrival of frontier generative models, however, has exposed the structural fragility of that approach in ways that are difficult to overstate. Models now surpass human baselines on established benchmarks so quickly that the tests themselves become obsolete within months of release, giving rise to what has been widely described as the "Evaluation Crisis" [1].

What makes this crisis genuinely consequential, beyond the methodological inconvenience it creates for benchmark designers, is the downstream effect it has on safety verification, alignment assurance, and commercial decision-making. When an organization cannot reliably determine whether a model is genuinely capable of the tasks it is being procured to perform—or whether its apparent performance reflects contaminated training data

Meta, USA

and leaderboard-targeted fine-tuning—the entire basis for responsible deployment becomes uncertain. The problem becomes especially acute in vertically specialized deployment environments: a hospital system integrating an LLM into clinical documentation workflows or a law firm using a generative model for contract review refuses to tolerate performance claims built on saturated or contaminated benchmarks because the consequences of misplaced confidence in those environments can be material and irreversible.

There is also a growing and well-documented divergence between the capability picture painted by published benchmark scores and the reality encountered by practitioners operating these systems in production. Engineers and researchers who deploy frontier models regularly observe performance gaps that published evaluations simply do not predict, a discrepancy that has eroded institutional trust in the evaluation ecosystem and created demand for measurement approaches that are more deeply tethered to real-world task distributions. Addressing that demand requires more than incremental improvements to existing benchmark design—it requires rethinking the entire architecture of how AI capability is defined, measured, and reported across a model's full deployment lifecycle.

1.1 The Phenomenon of Benchmark Saturation

Among the specific failure modes driving the evaluation crisis, benchmark saturation is perhaps the most visible and the most immediately damaging to the field's ability to distinguish genuinely capable systems from those that have simply been optimized to score well on a fixed set of problems. The historical arc here is instructive: foundational benchmarks like MNIST and Switchboard took more than two decades of sustained effort before models achieved human-level performance on them, providing the field with a long window of meaningful signal [2]. The contrast with contemporary practice could hardly be starker.

The 2025 Stanford AI Index documented performance gains on flagship benchmarks that would have been considered remarkable over a multi-year period but were achieved within a single year: an 18.8 percentage point improvement on the Massive Multi-discipline Multimodal Understanding (MMMU) benchmark and a 48.9 percentage point gain on the Google-Proof Q&A (GPQA) benchmark [3]. What this compressed timeline means in practice is that frontier models are now regularly clustering near the performance ceiling of nearly every static benchmark that exists, rendering those tests largely useless as instruments of meaningful differentiation. The distinction between a model that achieves 95% on a saturated benchmark and one that achieves 88% frequently reflects noise and dataset-specific artifacts rather than any genuine gap in operational capability [4].

The implications of this saturation problem extend well beyond the inconvenience it creates for academic leaderboard maintenance. In domain-specific deployment contexts—biomedical text synthesis, financial risk summarization, and legal reasoning—the narrow general-purpose benchmarks that populate most leaderboards provide no useful signal about the depth of specialized reasoning that practitioners actually need. Saturation in these general-purpose tests does nothing to reveal whether a model can correctly interpret a clinical trial protocol or identify ambiguous indemnification language in a commercial contract. This gap between the measurement instruments that exist and the evaluation needs that practitioners have is what makes saturation not merely a statistical annoyance but a structural obstacle to responsible AI procurement.

1.2 Data Contamination and the Illusion of Capability

If benchmark saturation is the most visible symptom of the evaluation crisis, data contamination—also referred to as benchmark leakage—is arguably the most insidious, because it operates silently and produces the appearance of capability where genuine generalization may be largely absent. The mechanism is straightforward: modern LLMs are trained on massive, largely unfiltered internet scrapes such as CommonCrawl, and because those datasets are broad enough to encompass much of the publicly accessible web, they almost certainly contain the very test sets that are later used to evaluate the models trained on them.

When a model encounters during evaluation a question it has effectively memorized during pre-training, the high score it achieves on that question is a measure of retrieval, not reasoning. Investigations into this problem have found that several prominent open-weight models reproduce verbatim examples from rigorous mathematics benchmarks, including MATH and GSM8k—a

finding that raises serious questions about how much of the impressive performance on those benchmarks reflects actual mathematical competence versus sophisticated memorization [2]. Longitudinal analyses have produced similarly troubling results, showing that model performance drops substantially on uncontaminated classification tasks released after a model's training cut-off date, while remaining inflated on older benchmarks that were almost certainly ingested during pre-training [2].

Detecting contamination at scale is technically non-trivial and has spawned its own methodological literature. Membership inference attacks, n-gram overlap detection, and perplexity-based diagnostic probes each offer partial coverage, but none is comprehensive, and none adequately addresses the subtler problem of indirect contamination—cases where a model has not memorized a specific test item verbatim but has been trained on enough structurally identical reasoning patterns that its performance on the test reflects something closer to pattern matching than novel inference. This form of contamination is particularly resistant to standard detection methods and may account for a substantial but currently unmeasurable share of inflated benchmark scores across the field. Building evaluation architectures that treat contamination resistance as a first-class design requirement, rather than a correction applied after the fact, is therefore not a luxury but a necessity.

1.3 Overfitting and Goodhart's Law in AI

The commercial pressures that accompany frontier model development have introduced a third and equally corrosive dynamic into the evaluation landscape: systematic overfitting driven by the incentive to perform well on public leaderboards. As benchmark rankings have become proxies for market positioning and funding narratives, developers have increasingly found it rational to optimize models specifically for benchmark performance, including through the generation of synthetic training data deliberately constructed to mimic the distributional properties of popular test sets.

The theoretical label for this dynamic is well established—it is a straightforward instance of Goodhart's Law, the principle that any measure that becomes a target ceases to function as a reliable measure—but the practical consequences in the AI context are serious. Evidence from late 2024 indicated that multiple model families suffered performance drops of approximately 10% when evaluated on GSM1k, a freshly generated and uncontaminated mathematics dataset, compared to their declared scores on the widely used GSM8k benchmark [5]. That gap is not explained by any genuine difference in the difficulty of the two datasets; it is explained by the fact that those models had been tuned, directly or indirectly, against the distribution of GSM8k in ways that did not improve their underlying mathematical reasoning. The 10% gap shows that public leaderboard numbers are not neutral measurements of these models' abilities; they are, to a large extent, artifacts of how those models were built to score well on the tests used to assess them.

2. The Paradigm Shift: LLM-as-a-Judge

Once static benchmarks began failing as reliable instruments, the field had little choice but to look for something structurally different—and what emerged, somewhat naturally, was the idea of turning the models themselves into evaluators. Rather than scoring outputs against a fixed key, the LLM-as-a-Judge approach puts a capable language model in the role of assessor, tasking it with judging another model's outputs against an explicitly defined rubric. This approach, widely referred to as LLM-as-a-Judge, has gained traction rapidly because it offers a scalable, flexible alternative to human evaluation without requiring the kind of fixed test sets that are vulnerable to saturation and contamination.

2.1 Scalability and Alignment with Human Preference

The gold standard for evaluating open-ended generative tasks has always been human judgment, but the economics and logistics of human evaluation make it unscalable at the pace and volume that modern model development demands. Evaluating a single fine-tuning run might require scoring thousands of output pairs across multiple task categories, and doing that with human annotators introduces delays, costs, and inter-annotator reliability challenges that compound at scale. The LLM-as-a-Judge framework sidesteps these bottlenecks by automating the evaluation process while retaining meaningful alignment with human preferences.

The empirical foundation for that alignment is strongest in the MT-Bench results, which demonstrated that capable LLM judges can match both controlled and crowdsourced human

preference judgments with agreement rates exceeding 80% [6]. Platforms such as the LMSYS Chatbot Arena have operationalized this alignment by combining crowdsourced human Elo ratings with automated judge pipelines to maintain leaderboards that are considerably more robust than purely static benchmark rankings [6]. Beyond leaderboard maintenance, the scalability of automated judges carries particular weight in iterative fine-tuning workflows, where each training cycle may generate large volumes of candidate outputs requiring immediate scoring. In these contexts, the latency and cost profile of an automated judge directly shapes the pace at which alignment work can proceed, making judge reliability and calibration a practical determinant of development velocity.

2.2 Known Biases in Automated Judges

The scalability advantages of LLM-as-a-Judge come with a well-documented set of limitations that must be confronted seriously before automated evaluation scores can be treated as reliable ground truth. When an AI system is asked to evaluate another AI system, the scoring process inevitably imports the biases embedded in the evaluating model, and several of those biases are systematic enough to meaningfully distort evaluation outcomes if left unaddressed [7].

Bias type	Description	Mitigation strategy
Position bias	The judge favours the first option in a pairwise comparison regardless of actual quality.	Swap output order and average scores; use position-agnostic prompting.
Verbosity bias	Longer, more elaborate answers are rated higher even when a concise response is more accurate.	Instruct the judge to penalise unnecessary length and reward conciseness in the rubric.
Self-enhancement bias	A judge disproportionately favours outputs from its own model family.	Use a diverse, heterogeneous panel of judges drawn from different developers.
Style / tone bias	A particular corporate writing style is preferred over substantive factual accuracy.	Decouple style from substance in the rubric; deploy factual verification sub-agents.

Table 1: Known Biases in Automated LLM Judges

These biases are not merely statistical noise—they encode structural preferences capable of systematically skewing reward signals during training and distorting capability assessments during post-deployment auditing. Addressing them effectively requires more than prompt-level instructions to the judge; it requires systematic calibration procedures that benchmark the judge's scoring distributions against held-out human expert ratings across diverse task types, output lengths, and stylistic registers. A judge who cannot demonstrate consistent scoring behavior across those dimensions is an unreliable measurement instrument, regardless of how impressive its general capabilities may be.

3. Advanced Evaluation Architectures: Multi-Agent Juries

Even after calibration efforts and carefully constructed prompting strategies, a single LLM judge carries scoring tendencies that are baked into its weights—preferences for certain styles, lengths, or model lineages that no amount of rubric engineering fully neutralizes. That limitation has driven the development of evaluation setups where the judgment is distributed across a panel of diverse models rather than concentrated in one, producing assessments that are harder to game and

more reliably correlated with genuine output quality.

3.1 The LLM-as-a-Jury Framework

The multi-agent jury approach, exemplified by frameworks such as CollabEval, draws a deliberate analogy to the human judicial system, distributing the evaluation task across a panel of heterogeneous models rather than concentrating it in a single evaluator. The operational logic of a typical multi-agent architecture unfolds across three phases that together address the principal weaknesses of single-judge evaluation [8]:

1. **Initial Independent Evaluation:** Multiple heterogeneous models independently score the output based on a shared rubric.
2. **Multi-Round Debate:** Each agent's initial rationale is shared with the others, and disagreements are worked through across successive rounds—a "critic" agent, for instance, might flag a hallucinated factual claim that the primary judge overlooked because it was distracted by the length and fluency of the response.
3. **Consensus / Final Judgment:** The agents converge on a final score, or a meta-judge aggregates the debated perspectives into a highly calibrated final rating [8].

Performance criterion	Single LLM judge	Multi-agent jury
Evaluation speed	Fast; low latency per scoring request.	Slower; multi-round debate adds coordination overhead.
Bias susceptibility	High; inherits all biases of the single model.	Reduced; heterogeneous panel suppresses systematic skew.
Accuracy vs. human consensus	Moderate; ~80% agreement on standard tasks.	Higher; debate rounds surface errors the primary judge missed.
Cost efficiency	Lower cost; single API call per evaluation.	Higher cost; multiple model calls per evaluation cycle.
Recommended use case	Routine, high-volume, low-stakes assessments.	Borderline outputs, safety decisions, calibration audits.

Table 2: Single LLM Judge vs. Multi-Agent Jury

The composition of the panel is itself a critical design variable—juries assembled exclusively from models sharing a common developer lineage reproduce the self-enhancement bias at the collective level, while heterogeneous panels drawn from architecturally distinct model families exhibit substantially reduced systematic scoring skew. The structured debate mechanism contributes additional calibration value by requiring each agent to articulate and defend its scoring rationale, a process that naturally surfaces inconsistencies and suppresses overconfident heuristic judgments that might otherwise pass unchallenged in a single-judge workflow.

From a practical standpoint, the orchestration overhead of multi-agent juries introduces latency and cost considerations that constrain their deployment at scale. Operational systems typically address these challenges through tiered evaluation architectures, where a single fast judge handles routine low-stakes assessments and the multi-agent jury is reserved for borderline outputs, high-consequence safety decisions, and periodic calibration audits. This hybrid structure preserves

the scalability advantage of automated evaluation while concentrating the highest rigor precisely where evaluation errors carry the greatest risk.

3.2 Reward Models and RLHF Metrology

The role of automated judges in AI evaluation extends well beyond post-training benchmarking into the training process itself. In Reinforcement Learning from Human Feedback (RLHF), the reward model serves as an embedded automated judge whose scoring signals directly shape the optimization trajectory of the primary LLM—which means that the quality of the reward model is not merely an evaluation concern but a determinant of the ultimate alignment properties of the system being trained.

Evaluating reward models rigorously requires going beyond the traditional NLP evaluation metrics—BLEU, ROUGE, and their variants—that were designed for different measurement purposes and are poorly suited to capturing preference alignment. Frameworks like the Preference Prediction Evaluation (PPE) benchmark fill this gap by evaluating reward models across multiple simultaneous dimensions, ensuring they consistently select human-preferred outputs without succumbing to reward hacking [9]. Reward hacking—the phenomenon in which the policy model learns to exploit systematic errors in the reward model's scoring rather than improving the genuine quality of its outputs—represents one of the most consequential failure modes in contemporary alignment pipelines. Its effects are particularly dangerous precisely because they tend to produce models that are superficially polished and score well on standard evaluations while harboring substantive reliability problems that only surface in deployment.

4. Dynamic Evaluation and Contamination Resistance

The data contamination problem described in the introduction does not yield to modest reforms in how static benchmarks are maintained—it requires a fundamental shift in the evaluation paradigm itself, from fixed datasets that can be ingested and memorized during pre-training toward dynamic evaluation targets that are continuously regenerated and structurally resistant to prior exposure.

4.1 Static-to-Dynamic Transformation

Dynamic evaluation addresses contamination at its root by ensuring that the evaluation surface is never stable long enough for a model to have encountered it during training. Two approaches have proven especially useful in tackling this:

- **Perturbation-based Evaluation:** Rather than leaving benchmark questions in their original form, perturbation engines rewrite the surface phrasing of each item—swapping vocabulary, restructuring syntax, and paraphrasing the setup—while deliberately keeping the underlying logic and the reasoning chain required to reach the correct answer intact. When a model handles the original question confidently but stumbles on the reworded version, that gap is fairly difficult to explain as anything other than memorization; a model that genuinely understood the concept would navigate both versions with comparable ease [10]. Effective perturbation must operate simultaneously at the lexical, syntactic, and semantic levels to ensure that memorized surface patterns cannot serve as proxy cues for the correct answer, and the integrity of each perturbation must itself be verified to confirm that difficulty and inferential structure have been faithfully preserved rather than inadvertently altered.
- **Temporal Filtering:** By continuously generating evaluation datasets drawn from sources published after a target model's training cut-off—news articles, code commits, scientific preprints, regulatory filings—it becomes possible to construct evaluation environments where contamination is structurally impossible rather than merely unlikely [10]. For models deployed in knowledge-intensive professional domains, this approach also provides a live signal of the model's capability on genuinely novel information, which is often precisely what practitioners most need to assess.

Evaluation criterion	Static benchmarks	Dynamic evaluation
Contamination risk	High; test sets often ingested during pre-training.	Low to none; tasks generated after training cut-off.
Benchmark longevity	Short; saturated within months of publication.	Sustained; continuously refreshed task pools.
Adaptability to new models	Limited; fixed content cannot track capability growth.	High; procedural generation scales with model advancement.
Measurement reliability	Declining; overfitting inflates reported scores.	Strong; novel tasks resist targeted optimisation.
Infrastructure requirement	Minimal; static files distributed once.	Significant; requires perturbation engines and temporal scrapers.

Table 3: Static vs. Dynamic Evaluation Frameworks

The transition to dynamic evaluation introduces new infrastructure requirements that the field is only beginning to address systematically. Because a dynamic benchmark by definition changes over time, published evaluation results must now carry richer metadata—benchmark version, generation timestamp, and model training cut-off date—to remain interpretable and reproducible. Establishing community norms around this metadata is a precondition for dynamic evaluation results to be meaningfully compared across studies and model generations.

4.2 Procedural Generation of Benchmarks

The most thoroughgoing solution to contamination is the procedural generation of evaluation tasks at inference time rather than the maintenance of any fixed corpus, however frequently refreshed. In mathematically and computationally well-defined domains such as coding and formal reasoning, automated frameworks can generate novel problems of arbitrary complexity with verified solution paths, guaranteeing that the exact problem instances used for evaluation have never existed anywhere on the internet and therefore cannot have been memorized during pre-training.

This approach introduces its own calibration challenges, since the difficulty distribution of procedurally generated tasks must be carefully controlled to allow meaningful comparison across evaluation cycles. Item Response Theory (IRT) modeling provides a principled mechanism for this calibration: by measuring how a reference population of models performs on generated items, it becomes possible to estimate difficulty parameters for those items and construct normalized scoring schemes that remain interpretable across dynamically generated task pools—preserving the comparative value of evaluation results even as the specific content of those evaluations changes continuously.

5. Multi-Dimensional and Safety Evaluation

Capability assessment, while critically important, captures only one dimension of what an AI system needs to demonstrate before it is trustworthy in high-stakes deployment contexts. As generative models become more agentic—making decisions, initiating actions, and operating with increasing autonomy across extended task horizons—evaluating their safety properties, ethical alignment, and behavioral consistency across demographic groups demands evaluation frameworks of considerably greater sophistication than those adequate for measuring task performance alone.

5.1 Automated Red Teaming

Safety evaluation in AI has historically relied on manual red teaming: structured adversarial testing by human experts designed to probe for conditions under which a system bypasses its safety constraints and produces harmful outputs. Manual red teaming remains valuable for the qualitative depth of understanding it generates, but it cannot scale to the volumes required by systems that generate millions of tokens per minute and are deployed across thousands of concurrent use cases simultaneously.

Automated red teaming addresses this scaling problem by using LLM judges specifically configured as adversarial agents—systems that autonomously generate large volumes of jailbreak

attempts, indirect prompt injections, and context manipulation sequences, while a separate evaluation layer assesses whether the target model successfully maintained its safety posture or produced a response that violated its intended constraints. The 2025 Singapore AI Safety Red Teaming Challenge, among other frameworks, demonstrated the practical value of these automated adversarial loops in surfacing vulnerabilities that manual testing would not have reached within any reasonable time budget [11].

The sophistication of adversarial agents in these frameworks has advanced considerably beyond the template-based prompt injection methods that characterized early red teaming systems. Contemporary automated red teaming employs multi-turn dialogue sequences that progressively escalate in adversarial intent, role-play framings that exploit a model's instruction-following tendencies rather than attacking its explicit safety filters directly, and indirect jailbreaks that embed harmful intent within apparently benign contextual frames. The continuing evolution of these attack strategies means that red teaming cannot be treated as a one-time pre-deployment gate; it must function as an ongoing operational practice that keeps pace with changes in both model capability and the adversarial landscape.

5.2 Multi-Dimensional Ethical Auditing

Reducing the ethical alignment of an AI system to a single scalar score is, at best, a convenient simplification and, at worst, a source of genuine governance failure. A model that scores impressively on a fairness metric while exhibiting brittle robustness under adversarial inputs or that produces well-calibrated explanations while generating a high rate of factual hallucinations is untrustworthy—yet a single-score evaluation regime would miss either of those failures. Advanced frameworks, such as the LLM Ethics Benchmark, explicitly reject this reductionism through a three-dimensional assessment architecture that evaluates moral reasoning across independent dimensions [12].

Evaluation axis	Primary objective	Measurement methodology
Fairness	Ensure outputs do not exhibit demographic bias or disparate impact.	Counterfactual fairness testing; demographic perturbation in prompts.
Robustness	Verify stability against adversarial attacks and prompt injections.	Automated generation of adversarial suffixes and semantic drift tests.
Explainability	Assess the model's ability to articulate its reasoning steps accurately.	Chain-of-Thought (CoT) verification using formal logic solvers.
Veracity	Quantify hallucination rates and factual accuracy across domains.	Retrieval-augmented LLM judges verifying claims against trusted databases.

Table 4: Multi-Dimensional Ethical Auditing Framework

Fairness is worth dwelling on specifically, because the deployment environments where AI systems are now operating make the stakes of getting it wrong very concrete. When a model is embedded in a credit scoring workflow, a hospital triage system, or a hiring platform, uneven performance across demographic groups does not stay confined to an evaluation report—it shows up as differential outcomes for real people, carrying legal exposure and causing harms that are difficult to walk back after the fact. Counterfactual fairness testing attempts to address this issue by keeping all variables in a prompt constant, except for the subject's demographic identity, and then measuring the extent to which the model's output actually changes in response to that single variable. How useful that method turns out to be depends heavily on how wide a net the testing framework casts: a narrow set of demographic variations will produce a clean-looking fairness score while leaving untested configurations—less frequently examined

group combinations, intersectional identities, and regional demographic framings—where meaningful disparities might still be sitting undetected.

Conclusion

What the current "Evaluation Crisis" represents, at its core, is a fundamental misalignment between the pace at which AI capabilities are advancing and the sophistication of the instruments being used to measure them. Static benchmarks that once served as useful proxies for model capability have been rendered unreliable by a combination of contamination, overfitting, and saturation, leaving practitioners, policymakers, and researchers with measurement tools that systematically overstate genuine capability and obscure the specific failure modes that matter most in high-stakes deployment contexts.

The case made across this brief is that addressing this misalignment is not a secondary concern to be resolved after the next capability frontier is reached but a precondition for responsible progress. Moving from static test sets to dynamic, contamination-resistant evaluation regimes, evolving from single-judge pipelines to carefully calibrated multi-agent juries, integrating multi-dimensional ethical auditing with continuous automated red teaming, and building evaluation infrastructure that can support regulatory compliance at scale—these are not incremental improvements to the existing paradigm. They represent a re-architecture of AI metrology appropriate to the moment the field is actually in. Without that re-architecture, the alignment and safety properties of deployed AI systems will remain genuinely unmeasurable, and the confidence expressed in them will rest on a foundation that closer examination reveals to be largely illusory.

References

[1] Jarosław Wasowski, "Is AI Cheating on the Test: Data Contamination, Gaming, and the Benchmark Crisis," Medium, 2026. Available: https://medium.com/@wasowski.jarek/mmlu-85-simpleqa-3-how-to-actually-evaluate-ai-models-in-2026-9dff2fba494f

[2] Sebastian Ruder, "The Evolving Landscape of LLM Evaluation," ruder.io, 2024. Available: https://www.ruder.io/the-evolving-landscape-of-llm-evaluation/

[3] Stanford University. "The 2025 AI Index Report." HAI, 2025. Available: https://hai.stanford.edu/ai-index/2025-ai-index-report

[4] Mubashara Akhtar, et al., "When AI Benchmarks Plateau: A Systematic Study of Benchmark Saturation," arXiv, 2026. Available: https://arxiv.org/pdf/2602.16763

[5] Hugh Zhang, et al. "A Careful Examination of Large Language Model Performance on Grade School Arithmetic," arXiv, 2024. Available: https://arxiv.org/pdf/2405.00332

[6] Lianmin Zheng, et al., "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," arXiv, 2023. Available: https://arxiv.org/pdf/2306.05685

[7] Jiayi Ye, et al., "Justice or prejudice? Quantifying Biases in LLM-as-a-Judge," arXiv, 2024. Available: https://arxiv.org/html/2410.02736v1

[8] Yiyue Qian, et al., "Enhancing LLM-as-a-Judge via Multi-Agent Collaboration," Amazon Science, 2025. Available: https://assets.amazon.science/48/5d/20927f094559a4465916e28f41b5/enhancing-llm-as-a-judge-via-multi-agent-collaboration.pdf

[9] Evan Frick, et al., "How to Evaluate Reward Models for RLHF," arXiv, 2024. Available: https://arxiv.org/pdf/2410.14872

[10] Simin Chen, et al. "Benchmarking Large Language Models Under Data Contamination: A Survey from Static to Dynamic Evaluation," ACL Anthology, 2025. Available: https://aclanthology.org/2025.emnlp-main.511.pdf

[11] Digital Policy Alert, "Singapore: Minister for Digital Development and Information released AI safety red teaming evaluation report," 2025. Available: https://digitalpolicyalert.org/event/27047-singapore-published-the-ai-safety-red-teaming-challenge-evaluation-report-2025

[12] Junfeng Jiao, et al. "LLM Ethics Benchmark: A Three-Dimensional Assessment System for Evaluating Moral Reasoning in Large Language Models," arXiv, 2025. Available: https://arxiv.org/pdf/2505.00853