PS-HalluBench: Taxonomy-Driven Benchmarking of LLM Hallucinations in Power Systems

S. M. Monzurul Hoque Chowdhury, Nabila Tasfiha Rahman, Qinghua Li
University of Arkansas, Fayetteville, AR, USA

arXiv:2611.22012v1 [eess.SY] 16 Nov 2026

***Abstract*—Large language models (LLMs) are increasingly explored for power-system applications, yet their reliability in engineering analysis remains uncertain. We introduce PS-HalluBench, a manually constructed benchmark of 200 questions that evaluates hallucination-related failures across five power-system task categories: numerical reasoning, network topology, constraint verification, false-premise acceptance, and abstention. We evaluate ten LLMs over the benchmark, yielding 39.35% overall correctness, with constraint verification achieving only 21.75%, and highlighting persistent difficulties in engineering reasoning. We further investigate engineering-grounded reasoning through numerical solver support and verified network information both of which show significant performance improvement. The findings reveal important limitations of standalone LLM reasoning and motivate integrating LLMs with verified engineering tools for reliable power-system analysis.**

***Index Terms*—Large language models, hallucination, benchmarking, power systems, reliability assessment.**

I. Introduction

Power-system engineering requires accurate knowledge of network topology, reliable system measurements, precise numerical calculations, and strict adherence to operational constraints. However, despite generating technically plausible responses, large language models (LLMs) may incorrectly assume network connections, produce inaccurate numerical solutions, recommend actions that violate operating limits, or provide confident answers when essential information is unavailable. Existing power-system LLM benchmarks primarily evaluate general capabilities, motivating a systematic, domain-specific assessment of hallucination-related engineering failures.

To address this gap, we introduce PS-HalluBench with the following contributions:

- To the best of our knowledge, it is the first dedicated LLM hallucination benchmark for power systems, comprising 200 manually constructed, reference-verified questions across five task-specific categories.
- A structured evaluation of ten LLMs across 2,000 independently graded responses, revealing category-specific reliability limitations.
- A systematic investigation of engineering-grounded LLM reasoning through iterative AC power-flow solver feedback and verified network topology, demonstrating their role in improving the reliability of numerical and topology-related conclusions.

II. Related Work

Hallucination and calibration: Existing surveys [1], [2] investigate the causes and characteristics of LLM hallucinations. Benchmarks such as TruthfulQA [3] and HaluEval [4] evaluate factual reliability, while abstention studies [5], [6] examine model behavior when insufficient information is available. However, hallucinations in power-system applications extend beyond factual inaccuracies, potentially arising during numerical calculations, topology reasoning, and operational decision-making.

Evaluation of LLMs in Power Systems: ElecBench [7] evaluates power-grid dispatch capabilities, including recognition of fictional engineering concepts. PowerBench [8] and PowerAgentBench-SS [9] focus on retrieval, reasoning, and agentic engineering workflows. Unlike these capability- and workflow-oriented benchmarks, PS-HalluBench targets five hallucination-related failure categories, through manually constructed, reference-verified questions, incorporating both valid and invalid premises, answerable and unanswerable scenarios, and independent human assessment.

III. Hallucination Taxonomy and Scope

LLM hallucinations encompass factuality failures, where generated claims contradict verifiable facts, and faithfulness failures, where outputs are inconsistent with supplied information or reasoning context [2]. These distinctions are particularly relevant to power-system engineering, where model-generated conclusions must be grounded in network configurations, electrical quantities, physical relationships, and operating constraints. Hallucination-related failures can therefore manifest as incorrect numerical assertions, fabricated network connections, erroneous operating-limit assessments, or conclusions drawn from invalid or insufficient information.

To systematically evaluate these manifestations, PS-HalluBench organizes power-system analysis into five categories: numerical reasoning, network topology, constraint verification, false-premise acceptance, and abstention. Numerical and constraint analyses primarily assess engineering correctness against verified solutions. Topology analysis evaluates the accuracy of network connectivity and consistency with supplied configurations, while false-premise analysis examines whether models recognize invalid engineering assumptions. Abstention analysis assesses whether models identify insufficient evidence and avoid unsupported conclusions. Together, these categories provide a taxonomy-driven evaluation of hallucination-related

failures through strict correctness against independently verified references.

Fig. 1 presents sample benchmark questions from each category, abbreviated for readability and due to the space limitation, together with failure identification procedure.

Numerical Hallucination-Related Errors: This category contains 40 questions involving power-system state estimation, DG placement, economic dispatch, and optimization. It evaluates the correctness of computed electrical quantities or optimal decisions against independently verified engineering references. Numerical operating-point answers were assessed using a 2% tolerance, while DG-placement answers required the exact reference bus set.

Topology Hallucinations: This category has 40 questions evaluating network connectivity, branch relationships, and system configurations. Questions involve both explicitly supplied network information and standard IEEE test systems. Responses are checked against verified network connectivity and switching states. Incorrect connections, omitted branches, or unsupported topology assumptions constitute failures.

Constraint-Violation Failures: This category comprises 40 questions requiring LLMs to identify violations of voltage, branch-loading, and generator reactive-power limits under specified operating conditions, including post-contingency scenarios. AC power-flow calculations may be required to determine the resulting electrical state, but no optimization is involved. Unlike numerical tasks, responses are evaluated on the identified violations and affected components. Missed or falsely reported violations, incorrect components, or incorrect violation types constitute failures.

False-Premise Acceptance: This category contains 40 questions designed to evaluate whether LLMs appropriately assess the validity of engineering assumptions. The questions include both valid and invalid premises, requiring models to identify contradictions when present while avoiding unjustified rejection of valid assumptions.

Abstention and Evidence Insufficiency: This category contains 40 questions evaluating whether LLMs can distinguish between sufficient and insufficient engineering information. Questions include both answerable and unanswerable scenarios, requiring models to abstain when essential evidence is missing while providing valid conclusions when sufficient information is available.

The categories distinguish the engineering contexts where hallucination-related failures are evaluated, rather than assigning mutually exclusive causes to individual errors.

IV. Benchmark Construction and Ground Truth

The benchmark consists of 200 manually constructed questions. Unlike ElecBench [7], which employed GPT-4-assisted question generation, our questions were developed manually to reduce potential biases arising from model-generated assumptions and reasoning patterns [10]. This approach enables deliberate construction of challenging engineering scenarios targeting specific hallucination-related failures. The questions were intentionally designed to create a high evaluation standard for current and future LLMs, rather than primarily assessing basic domain knowledge.

Each question was accompanied by an independently verified reference answer. Reference answers were independently prepared by two authors and subsequently cross-checked. For numerical and constraint analyses, dedicated Python solvers were used to compute reference solutions, with convergence, mathematical consistency, and applicable operating constraints checked to validate the results. Topology answers were verified against the corresponding network configurations and specified switching states. For false-premise and abstention tasks, reference answers were established based on the supplied evidence and the validity or sufficiency of the underlying engineering information.

The benchmark is released on Github [11]. The repository provides benchmark questions, category-specific grading rubrics, prompting and inference protocols, and model configurations to support transparency and reproducibility.

V. Experimental Procedure

A. Models and Evaluation Criteria

We evaluate ten LLMs: GPT-5.4, GPT-5.1, GPT-4o, Qwen3.5-27B, Qwen3.5-9B, DeepSeek-R1-Distill-Qwen-7B, Gemma-3-12B-IT, Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct, and Mistral-Nemo-Instruct-2407. Each model was evaluated on all 200 benchmark questions, yielding 2,000 model-question evaluations. Open-weight models were deployed on NVIDIA A100 GPUs at the Arkansas High Performance Computing Center, while GPT models were accessed through the OpenAI Responses API.

All models were evaluated without external tools, code execution, retrieval, or numerical solvers. Local models and GPT-4o used temperature 0.7 and top-p 0.8, with thinking mode enabled for Qwen3.5. GPT-5.1 and GPT-5.4 used default sampling settings, along with medium reasoning effort. No explicit instructions encouraged models to abstain or challenge question premises, allowing us to assess whether they independently recognized insufficient evidence or invalid assumptions. Only final answers, excluding reasoning traces, were graded.

Initial trials with 2,000 and 8,000 output tokens resulted in truncations. The final evaluation therefore used a maximum of 16,384 output tokens, with no truncated responses among the 2,000 evaluations. For OpenAI reasoning models, this budget includes internal reasoning and visible output tokens. Results reflect single-sample performance under model-dependent inference configurations; generation variability was not separately analyzed.

Two authors independently graded all 2,000 anonymized responses using predefined category-specific criteria, blinded to model identity. Initial agreement was 97.85% (1,957/2,000), with Cohen's $\kappa = 0.9523$. Category-wise agreement was 99.25% for numerical, 98.25% for topology, 98.75% for constraints, 96.25% for false-premise, and 96.75% for abstention tasks. The 43 disagreements were resolved through discussion.

For each category, strict correctness is defined as

Failure Mode	Representative Question (Abbreviated)	How Failure Is Identified
NUMERICAL	For a synthetic 10-bus radial system, given branch parameters, bus loads, DG specifications, and operating limits, find the optimal single-DG placement.	The model must compute the requested electrical quantity or determine an optimal operating decision. Failure occurs when the result exceeds the accepted tolerance or the decision differs from the verified optimum.
TOPOLOGY	Given a 7-bus network with specified branch connections, which buses are directly connected to Bus 4?	The model must identify network configurations using supplied information or knowledge of standard IEEE systems. Failure occurs when its conclusions disagree with verified topology.
CONSTRAINTS	For a synthetic 6-bus AC system, given branch, load, generator, and voltage-limit specifications, identify voltage violations.	The model must identify violated operating limits under specified system conditions. Failure occurs when it misses violations, reports nonexistent ones, or identifies incorrect components.
FALSE PREMISE	In a 10-bus network, line 7–10 is outaged. An unreliable report suggests that Bus 7 is isolated from the main grid. Determine the temporary generation required at Bus 7.	The model must assess premise validity before drawing a conclusion. Failure occurs when it accepts an invalid premise, or rejects a valid one, and based on that, reaches an incorrect engineering conclusion.
ABSTENTION	Given a lossless line connecting buses with voltage magnitudes of 1.00 and 1.01 pu, determine the direction of active-power transfer.	The model must determine whether sufficient information exists to answer. Failure occurs when it provides an unsupported answer or unnecessarily abstains when a valid answer is possible.

Fig. 1. Representative abbreviated questions from the five PS-HalluBench failure categories and their operational failure criteria. Original questions include complete system specifications, operating conditions, and modeling assumptions as applicable.

$$A_c = \frac{1}{10|Q_c|} \sum_{m=1}^{10} \sum_{q \in Q_c} \mathbf{1}[y_{mq} = \text{correct}], \tag{1}$$

where Q_c denotes the questions in category c, and y_{mq} is the evaluated outcome of model m on question q. Strict correctness measures the reliability of engineering conclusions in hallucination-related task settings without attributing individual errors to specific underlying mechanisms.

For false-premise and abstention analyses, we additionally compute subtype-balanced strict correctness:

$$A_c^{\text{bal}} = \frac{1}{2}\left(A_{c,1} + A_{c,2}\right), \tag{2}$$

where $A_{c,1}$ and $A_{c,2}$ are strict correctness rates computed using (1) within each subtype. This assigns equal weight to both subtypes, regardless of their frequencies, while still requiring correct engineering conclusions.

We computed 95% percentile bootstrap confidence intervals using 50,000 question-level resamples. Within each category, all ten model outcomes associated with each sampled question were resampled together. Overall confidence intervals were computed using category-stratified resampling. These intervals characterize question-level variability rather than repeated-generation uncertainty.

B. Response Runtime Reporting

We report median response latency (t_{50}) across models and failure categories. Runtime measurements are presented descriptively, as API-based and locally deployed models operate under different computational environments and are not directly hardware-normalized.

VI. Results and Discussion

Overview of Category Performance: Across 2,000 evaluated responses, the models achieved 39.35% overall strict correctness (787/2,000), highlighting challenges in reliable power-system analysis. Performance varied across the five categories. Constraint verification exhibited the lowest correctness (21.75%), followed by topology reasoning (30.0%) and numerical analysis (31.0%). False-premise and abstention tasks achieved 55.25% and 58.75%, respectively. These findings indicate that numerical computation, network interpretation, and operating-limit verification remain particularly challenging. Even with sufficient generation capacity to produce responses for all questions, a considerable proportion of engineering conclusions remained incorrect. Fig. 2 presents model-specific correctness and runtimes, while Table I summarizes category-level performance.

TABLE I
Results by hallucination-related failure category.

Category	Correct	Acc.	Bal. Acc.	95% CI	t_{50} (s)
Numerical	124	31.00%	–	[23.8,38.3]	22.21
Topology	120	30.00%	–	[22.3,38.0]	5.13
Constraints	87	21.75%	–	[18.0,25.5]	12.79
False premise	221	55.25%	56.09%	[47.3,63.3]	16.18
Abstention	235	58.75%	59.66%	[49.5,67.8]	12.30
All	787	39.35%	–	[36.0,42.7]	–

Numerical Hallucination-Related Failures: Numerical tasks achieved 31.0% correctness (124/400), with a median response runtime of 22.21 s. GPT-5.4 achieved the highest accuracy (60.0%), followed by Qwen3.5-27B (55.0%) and GPT-5.1 (52.5%). The low overall correctness highlights persistent difficulties in multi-step calculations and optimization, even when extended reasoning is permitted.

Topology Hallucinations: Topology tasks achieved 30.0% correctness (120/400), with a median response runtime of 5.13 s. GPT-5.4 recorded the highest accuracy (70.0%), followed by GPT-5.1 (52.5%) and DeepSeek-R1-Distill-Qwen-7B (35.0%). Incorrectly recalled network configurations and unsupported connectivity assumptions contributed to observed failures.

Constraint Violations: Constraint-verification tasks

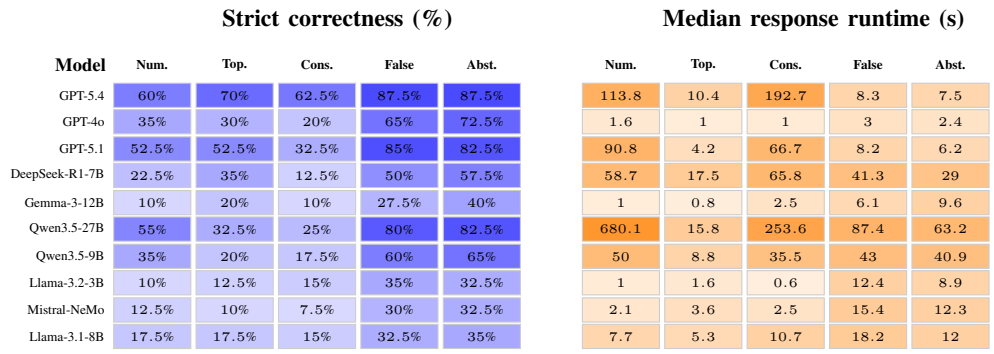


Fig. 2. Per-model and per-category results ($n = 40$ questions per cell). Left: strict correct-response rate. Right: median observed response runtime, with logarithmic shading. Categories are numerical, topology, constraints, false premise, and abstention. Individual 95% Wilson intervals are approximately ± 15 percentage points near 50% accuracy.

achieved 21.75% correctness (87/400), the lowest among all categories. GPT-5.4 performed best (62.5%), distantly followed by GPT-5.1 (32.5%) and Qwen3.5-27B (25.0%). Errors might arise from incorrect state estimation, violation assessment, or both. The findings emphasize the difficulty of reliable constraint verification using standalone LLMs.

False-Premise Acceptance: False-premise tasks achieved 55.25% strict correctness (221/400), with a median response runtime of 16.18 s. GPT-5.4 achieved the highest accuracy (87.5%), followed by GPT-5.1 (85.0%) and Qwen3.5-27B (80.0%). The category contains 32 invalid and 8 valid premises. Strict correctness was 54.69% (95% CI: 46.3–62.8%) for invalid premises and 57.50% (95% CI: 35.0–81.3%) for valid premises, yielding a subtype-balanced accuracy of 56.09%. Given the small valid-premise subset, the seen similarity does not establish equivalent performance across subtypes.

Abstention Failures: Abstention tasks achieved 58.75% strict correctness (235/400), with a median response time of 12.30 s. GPT-5.4 achieved the highest accuracy (87.5%), followed by GPT-5.1 and Qwen3.5-27B (82.5% each). The category contains 21 unanswerable and 19 answerable questions. Strict correctness was 41.43% (95% CI: 30.0–52.9%) for unanswerable cases and 77.89% (95% CI: 70.0–85.8%) for answerable cases, yielding a subtype-balanced accuracy of 59.66%. The 36.47% gap indicates substantially greater difficulty in recognizing insufficient engineering information than answering sufficiently specified questions.

Cross-Model Comparison and Discussion: The results reveal substantial differences in correctness across the evaluated models. GPT-5.4 achieved the highest overall accuracy (73.5%), followed by GPT-5.1 (61.0%) and Qwen3.5-27B (55.0%). GPT-5.4 also achieved the highest correctness across all five categories. Performance differences were not explained by parameter count alone. For example, DeepSeek-R1-Distill-Qwen-7B achieved 35.5% correctness, exceeding Gemma-3-12B (21.5%), while Qwen3.5-9B achieved 39.5%. These observations suggest that model architecture, reasoning configurations, and model training may influence engineering reliability, although their individual effects were not isolated.

Constraint verification remained particularly challenging, with even GPT-5.4 achieving only 62.5% correctness. However, based on its highest overall accuracy, GPT-5.4 was selected for studying the enhancement of LLM reliability next.

VII. Enhancing LLM Reliability in Power Systems

The benchmark results indicate that models such as GPT-5.4, GPT-5.1, and Qwen3.5-27B achieve comparatively higher correctness on false-premise and abstention tasks, while numerical analysis, constraint verification, and topology reasoning remain challenging. These findings motivate combining LLMs with verified engineering information and numerical tools rather than relying solely on model-generated conclusions. We investigate this approach through two GPT-5.4 experiments: DG-placement optimization with iterative AC power-flow feedback and topology reasoning with verified IEEE network information.

A. Numerical Analysis with Iterative Solver Support

To investigate solver-assisted numerical reasoning, we adapted the DG-placement problems from benchmark dataset into 40 standardized scenarios using synthetic radial distribution feeders (10–69 buses), with one to three DG units and search spaces ranging up to 2,278 candidate placement combinations. These scenarios constitute a separate controlled experiment rather than the original 40 numerical benchmark questions. The goal of this experiment is to evaluate whether iterative solver feedback improves LLM-guided engineering decisions. DG outputs were fixed, while locations were optimized. The objective was $J_0 = P_{\text{loss}} + 3\sum_i (|V_i| - 1)^2$, accounting for active power loss and voltage deviation. Hard-constrained cases enforced voltage and branch limits; soft-constrained cases penalized violations. Exhaustive search established reference optima.

The AC power-flow solver does not directly determine optimal DG locations. Instead, the solver evaluates candidate DG placements proposed by LLM, computing bus voltages, branch currents, power losses, and constraint violations. In the iterative framework, GPT-5.4 proposes placements, receives their solver-computed objectives and feasibility results, and uses this feedback to guide subsequent proposals. The final placement was selected as the best feasible candidate among the solver-evaluated configurations. Each method received at most 10 solver evaluations. GPT-5.4 used one, two, or five

TABLE II
DG-PLACEMENT PERFORMANCE ($n = 40$; MAX. 10 SOLVER CALLS).

Method	Correct	Acc.	Gap (%)	t_{50} (s)
GPT only	8/40	20%	9.41	146.62
GPT + solver (1 iteration)	17/40	42.5%	10.98	78.81
GPT + solver (2 iterations)	22/40	55.0%	3.81	176.92
GPT + solver (5 iterations)	28/40	70.0%	2.91	444.93
Random + solver	–	23.0%	8.92	0.025
Heuristic + solver	22/40	55.0%	21.76	0.024
Adaptive greedy + solver	14/40	35.0%	3.29	0.024

Gap is mean $100(J - J^*)/|J^*|$. Runtime is median wall-clock time.

TABLE III
GPT-5.4 TOPOLOGY ANALYSIS PERFORMANCE WITH AND WITHOUT VERIFIED NETWORK INFORMATION ($n = 40$).

Method	Correct	Accuracy	t_{50} (s)
GPT only	28/40	70.0%	10.41
Verified topology	39/40	97.5%	2.52

planning rounds, with budgets of 10, 5+5, and 2+2+2+2+2 evaluations, respectively.

To distinguish the contribution of solver access from that of LLM-guided selection, we included three solver-guided baselines: random search (20 repetitions), a topology/load-based heuristic, and adaptive greedy search. The heuristic ranked placements using downstream loads and bus depth, while adaptive greedy search iteratively selected candidates based on previous solver outputs. Five-round GPT achieved 70.0% accuracy, versus 42.5% with one round, and reduced mean optimality gap from 10.98% to 2.91%. The optimality gap measures relative objective degradation from the exhaustive optimum. These results suggest that iterative solver feedback improves GPT-guided search, although additional planning rounds increase inference latency.

B. Topology Reasoning with Verified Network Information

We examined GPT-5.4 topology reasoning with and without verified network information across the same 40 benchmark questions. The grounded condition provided authoritative connectivity data from standard IEEE test systems. As shown in Table III, correctness increased from 70.0% (28/40) to 97.5% (39/40). Verified information corrected 11 previously incorrect responses, with no regressions (exact McNemar test, $p < 0.001$). For questions requiring knowledge of standard IEEE systems, correctness increased from 63.3% (19/30) to 96.7% (29/30). For questions with supplied connectivity, correctness increased from 90.0% (9/10) to 100% (10/10). These findings suggest that authoritative network information can reduce errors associated with recalled system topology, supporting evidence-grounded LLM analysis.

C. Toward Reliable LLM Integration in Power Systems

Our findings suggest that reliable LLM integration in power systems requires more than improved reasoning capabilities. As observed in false-premise and abstention tasks, a single LLM may overlook critical information, accept incorrect assumptions, or fail to recognize insufficient evidence. These failures motivate agentic frameworks in which specialized agents independently verify engineering assumptions, assess evidence sufficiency, and evaluate proposed conclusions before they inform decisions. Furthermore, improvements from numerical solver feedback and verified network information suggest that such frameworks could incorporate authoritative engineering tools rather than rely solely on LLM-generated reasoning. This approach offers a potential direction for more reliable LLM-integrated power-system analysis and decision support.

VIII. CONCLUSION

PS-HalluBench evaluated ten LLMs across 200 questions covering five hallucination-related failure categories. Overall correctness was 39.35%, with particularly low performance in constraint verification, topology reasoning, and numerical analysis. These findings highlight limitations of standalone LLMs and motivate integration with numerical solvers, network information, and engineering verification mechanisms for reliable power-system decision support.

REFERENCES

[1] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 248:1–248:38, 2023.

[2] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, 2025.

[3] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2022, pp. 3214–3252.

[4] J. Li, X. Cheng, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, "HaluEval: A large-scale hallucination evaluation benchmark for large language models," in *Proc. Conf. Empirical Methods Natural Language Processing (EMNLP)*, 2023, pp. 6449–6464.

[5] N. Madhusudhan, S. T. Madhusudhan, V. Yadav, and M. Hashemi, "Do LLMs know when to NOT answer? investigating abstention abilities of large language models," in *Proc. 31st Int. Conf. Comput. Linguistics (COLING)*, 2025, pp. 9329–9345.

[6] P. Kirichenko, M. Ibrahim, K. Chaudhuri, and S. J. Bell, "AbstentionBench: Reasoning LLMs fail on unanswerable questions," *Adv. Neural Inf. Process. Syst.*, vol. 38, 2025.

[7] X. Zhou, H. Zhao, Y. Cheng, G. Liang, G. Liu, W. Liu, Y. Xu, and J. Zhao, "ElecBench: A power grid dispatch evaluation benchmark for large language models," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2025, pp. 1–5.

[8] X. Wang, Y. Yao, J. Ding, Y. Jiang, Z. Xu, Y. Jiang, J. Xu, and D. Cheng, "PowerBench: A benchmark for agentic retrieval and reasoning in power systems," *arXiv:2609.34492*, 2026.

[9] C. Mylonas, M. Foti, A. Pomarico, M. Duarte, Q. Zhang, and E. Varvarigos, "PowerAgentBench-SS: A benchmark for agentic AI in power system steady-state studies," *arXiv:2606.18789*, 2026.

[10] P. Yuan, Y. Li, S. Feng, X. Wang, Y. Zhang, J. Shi, C. Tan, B. Pan, Y. Hu, and K. Li, "Silencer: From discovery to mitigation of self-bias in LLM-as-benchmark-generator," in *Adv. Neural Inf. Process. Syst.*, vol. 38, 2025, pp. 129 636–129 658.

[11] S. M. M. H. Chowdhury, N. T. Rahman, and Q. Li, "PS-HalluBench:-benchmark-materials," Available: https://github.com/smshihab53-ux/PS-HalluBench-Benchmark-materials, 2026.