SMARTGRIDBENCH: COHORT-BOUND EVALUATION OF INDUSTRIAL LLM AGENTS FOR TRANSFORMER MAINTENANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Power-transformer maintenance requires an agent to combine telemetry, fault interpretation, forecasting, and work-order records before recommending an action. We introduce SmartGridBench, a synthetic transformer-maintenance extension of AssetOpsBench (Patel et al., 2025), and use it to separate tool-interface choices from planning choices. The frozen resource contains 36 structurally validated scenarios; the principal quality comparison evaluates 31 hand-authored scenarios, while a later repository expansion to 61 scenarios was not evaluated in these experiments. On the 31-scenario core, each configuration has five trials per scenario: direct tool calls receive 67/155 recorded judge passes, MCP Agent-as-Tool receives 57/155, and Verified Plan-Execute receives 86/155. A separate six-scenario latency cohort records median completion times of 8.51 s, 12.22 s, and 37.92 s for those three configurations; the cohorts are not pooled. A 21-case, outcome-stratified cross-family judge check agrees with the historical pass label on 13/21 cases (61.9%) and assigns 4 versus 10 passes, which demonstrates evaluator sensitivity rather than judge accuracy. The synthetic fleet fails an existing DGA realism check, and independent human and domain validation remain unavailable. We therefore present SmartGridBench as a reproducible fixture for conditional systems evaluation; establishing maintenance readiness or deployment safety would require further validation.

1 INTRODUCTION

A maintenance recommendation can be wrong even when every tool call succeeds. An agent might retrieve a transformer measurement, interpret the wrong sensor, and confidently recommend taking the asset out of service. Testing whether the software ran is therefore different from testing whether the recommendation follows from the evidence.

Power transformers provide a concrete setting for studying this distinction. We ask two narrow questions: how do the tested direct-call and Model Context Protocol (MCP; Anthropic, 2024) implementations differ in completion time and judged task quality under the same iterative workflow, and how do planning and verification change those outcomes when tools are accessed through MCP? We do not claim that these comparisons isolate a general protocol effect: several configurations bundle transport, serving, prompting, or client-environment changes.

Our contributions are a scoped synthetic benchmark extension, a cohort-bound comparison with explicit denominators, and an auditable account of judge and data-validity limits. The five invalid-input fixtures, later 25-scenario repository additions, and audit-derived taxonomy are kept separate from the evaluated quality denominator. The relationship to AssetOpsBench (Patel et al., 2025) is an extension of its industrial asset-maintenance setting, not an independent validation of that benchmark. MCP (Anthropic, 2024) supplies the tool-interface protocol and vLLM (Kwon et al., 2023) the principal serving system; neither reference establishes the validity of our synthetic tasks or historical comparisons.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 BENCHMARK AND SCOPE

2.1 RESOURCE CONSTRUCTION

The benchmark combines telemetry, fault and maintenance standards, forecasting, and work-order tools for synthetic transformer-maintenance tasks. The tracked processed tables describe a synthetic fleet. Source-derived data descriptions in the historical appendix do not match those tables; they cannot establish which inputs a historical run used without a run-specific data manifest. The shared identifier is a construction device, not evidence of measurements from the same real asset or a natural joint distribution.

The frozen resource contains 36 validated scenario files: 31 hand-authored scenarios and five generated candidates retained after manual disposition and edits. Ten other generated candidates were rejected as near-duplicates. Five negative schema fixtures are maintained separately to test invalid inputs; they are not additional evaluated maintenance tasks. The principal quality matrix uses the 31 hand-authored scenarios. A later repository expansion contains 61 scenarios, but its 25 additions were not evaluated in the frozen study.

Structural validation checks format, ranges, and consistency; it does not establish that a task represents real maintenance work. The existing statistical DGA check fails to match a labeled reference distribution, so the synthetic resource is suitable for specified repeatable fixtures rather than claims of distributional realism.

2.2 TASK AND CONFIGURATIONS

Scenario SGT-001 asks the agent to list all available sensors for transformer T-011 and include each latest timestamped value. More involved scenarios combine tool domains, but this retrieval example is not evidence of safe diagnosis.

2.3 AGENT CONFIGURATIONS

The iterative Agent-as-Tool baseline lets the language model choose tool calls and use their returned observations to continue the task. The direct-call configuration invokes the tools locally; the MCP configuration reaches the corresponding tool domains through MCP. Plan-Execute first constructs a plan and then executes its steps. Verified Plan-Execute adds an intermediate check of step results. Its verifier uses the actor's own language-model instance; it is self-verification, not an independent assessment. Self-Ask variants add explicit intermediate questions to the planning process.

We keep configuration identifiers in Table 1 to connect the comparisons to the stored run records, but use descriptive names in the text. The optimized rows are implementations to compare, not isolated measurements of a protocol primitive.

The principal serving system is vLLM (Kwon et al., 2023). The stored core configurations use FP16 actor serving. Although configuration C explicitly requests prefix caching, sampled historical startup logs also enable it in A and B. Thus C is not an established cache-off/cache-on contrast, and its difference from B cannot be attributed to prefix caching or transport alone. The serving follow-on D combines compressed-tensors INT8 weights, FP8 key/value caching, and prefix caching. These bundled settings motivate reporting the configuration as a whole rather than assigning its measured effect to an individual optimization.

The primary actor is Llama-3.1-8B-Instruct (Meta, 2024a). Hosted supplementary runs use Llama-3.3-70B-Instruct (Meta, 2024b) through IBM watsonx. The recorded outcome judge is `llama-4-maverick-17b-128e-instruct-fp8`, also hosted through watsonx; the underlying Maverick model is documented by Meta (Meta, 2025). These model references identify the models, not the effective settings of the historical hosted requests. Thus the outcome judge is a different checkpoint from either actor, although all belong to the Meta Llama family. That distinction corrects an ambiguity in the earlier presentation; it does not establish independence from model-family biases. The outcome judge and the in-loop verifier perform different jobs and must not be conflated.

Table 1: Configuration map. All planning rows use MCP. The serving follow-on does not appear in the 31-scenario quality matrix.

ID	Agent workflow	Tool/serving condition
A	Iterative Agent-as-Tool	Direct calls
B	Iterative Agent-as-Tool	MCP baseline
C	Iterative Agent-as-Tool	Optimized MCP configuration
D	Iterative Agent-as-Tool	Optimized MCP and serving
Y	Plan-Execute	MCP
YS	Plan-Execute with Self-Ask	MCP
Z	Verified Plan-Execute	MCP
ZS	Verified Plan-Execute with Self-Ask	MCP

2.4 What the outcome score measures

The outcome judge receives the task, its characteristic expected answer, the plan, a trajectory summary, and the final answer as a separate field. The recorded rubric has six equally weighted binary dimensions: task completion, data retrieval accuracy, result verification, agent sequence correctness, clarity and justification, and absence of hallucinations. The last dimension inverts the stored hallucination flag. The score is the fraction of favorable judgments. The historical pass threshold is 0.6, which requires four of six dimensions.

This rule has an important limitation: a trajectory can pass while failing task completion or containing a hallucination, provided enough other dimensions are favorable. We therefore call this outcome a *judge pass*, not task success or safe maintenance performance. The metric should not be used as an operational acceptance criterion.

The judge implementation uses temperature zero, a 512-token output limit, and prompt version `assetopsbench-6d-v1`. Its saved requests contain no explicit reasoning-mode setting; we found no recorded rationale for the historical output limit. The later Flash calibration provides a measured reason for its own non-thinking setting, not a retrospective explanation of the Maverick configuration. The historical pipeline truncates the supplied trajectory text to 8,000 characters. Reconstructing that input from the stored core records finds 100 of 1,085 trajectories above the limit (9.2%). The counts by configuration A, B, C, Y, YS, Z, and ZS are respectively 6, 9, 10, 10, 10, 30, and 25, each out of 155. The final-answer field is supplied separately; the truncation concerns the tool and intermediate-step record. Whether this uneven loss of evidence changes the full-grid comparison remains unresolved. The exploratory pilot described in the appendix repeats the scoring with full trajectories. Only three of its 28 sampled cases have an input that the cap actually changed, and one of those three changes its score. Separately, 5 of the 25 identical-input pairs return a different rubric score from the cross-family pilot judge despite temperature zero and byte-identical requests, one of them crossing the pass threshold; run-to-run variance in that judge is therefore of the same order as the truncation effect we can measure. The expected answer is visible to the judge: this is reference-based grading, not a blind assessment. Neither low temperature nor an answer reference establishes agreement with a human evaluator.

2.5 Cohorts and repeated trials

The main quality analysis joins the evidence registry to the score records for the core-15 and core-16 cohorts. Together these contain the same 31 scenario identifiers for each of seven configurations, with five recorded trials per scenario. The revision analysis checks membership, trial counts, and duplicate keys before comparing configurations. It does not fill missing results or select trials to improve agreement with the paper.

The latency analysis uses a separate six-scenario cohort with five trials per configuration. Timings from this cohort cannot be combined with the 31-scenario score matrix as if they came from identical workloads. The hosted 70B and mitigation experiments are separate cohorts as well. The reported 2,420 eligible score rows span 37 groups across the study; they are neither 2,420 distinct tasks nor one balanced experimental design.

The original environments are not perfectly matched. The vanilla Plan-Execute client uses Python 3.12, whereas the other principal clients use Python 3.11. We have not measured the isolated effect of that difference and do not dismiss it as negligible. Likewise, combined serving, transport, and prompting changes cannot support attribution to any one component without an ablation.

3 RESULTS

3.1 CORE QUALITY COHORT

The primary matrix contains 31 distinct hand-authored scenarios and five recorded trials per scenario (155 trials per configuration). The recorded judge means and passes are shown below. A judge pass is a thresholded output of the historical evaluator, not expert-certified operational success.

Table 2: Recorded outcomes on the 31-scenario core. Every row has 31 scenarios and five trials per scenario. Values are historical judge outputs, not safety or operational-success labels.

Configuration	Mean score	Judge passes
Direct Agent-as-Tool	0.474	67/155
MCP Agent-as-Tool	0.442	57/155
Optimized MCP	0.473	61/155
Plan-Execute	0.582	76/155
Plan-Execute + Self-Ask	0.606	78/155
Verified Plan-Execute	0.599	86/155
Verified PE + Self-Ask	0.620	80/155

The MCP iterative row has fewer recorded passes than direct calls in this matrix, while the verified planning row has the largest pass count and its Self-Ask variant has the largest mean. These are associations for the tested implementations, actor, prompts, and scenarios. They do not establish that MCP causes lower quality or that planning causes a reliable improvement. Post-hoc scenario-cluster intervals are exploratory and unadjusted; the direct-versus- MCP intervals include zero. Appendix D gives the resampling procedure and the full interval table.

3.2 SEPARATE LATENCY COHORT

Latency uses six scenarios and five trials per configuration (30 trials per row), not the 155-trial quality matrix. The direct, MCP, and Verified PE rows have recorded medians of 8.51 s, 12.22 s, and 37.92 s, with 95th percentiles of 20.73 s, 13.99 s, and 94.93 s, respectively.

Table 3: Completion times in the separate six-scenario cohort. Each row has 30 trials.

Configuration	Median (s)	95th percentile (s)	Passes
Direct Agent-as-Tool	8.51	20.73	6/30
MCP Agent-as-Tool	12.22	13.99	3/30
Optimized MCP	12.01	14.58	6/30
Plan-Execute	29.62	60.74	17/30
Verified Plan-Execute	37.92	94.93	22/30

These timings are cohort-specific. Prefix caching was enabled in purported baselines, follow-on rows bundle serving changes, and stored metadata does not independently identify the GPU. We therefore do not interpret the table as a causal protocol-overhead estimate or combine it with the Core31 quality rows. The table reports marginal completion-time quantiles, not quantiles of paired per-trial differences. Subtracting two medians would not estimate the median paired overhead.

3.3 SUPPLEMENTARY OBSERVATIONS AND JUDGE DIAGNOSTICS

Hosted 70B observations use a separate 15-scenario cohort with unequal replication: three trials per scenario for the iterative rows (45 trials) and five for the planning rows (75 trials). They are not

comparable term-by-term with the 31-scenario core. Verified Plan-Execute with Self-Ask has the strongest recorded supplementary mean, 0.644 with 39/75 passes. This is a serving-path observation, not cross-family actor replication.

On 21 cases held out from settings calibration (but previously seen in the exploratory pilot), the cross-family judge's first draw agrees with the historical pass label on 13/21 cases (61.9%). The alternative judge passes 4 cases versus 10 for the historical judge. The sample is outcome-stratified and not representative; the result measures evaluator sensitivity, not which judge is correct.

4 Validity and Limitations

The synthetic fleet does not reproduce the joint distribution of a real fleet, and its DGA realism check fails. Human-authored scenarios can reflect author assumptions, and independent domain labels are unavailable. The outcome is a six-dimension LLM-judge score with a historical 0.6 threshold; a judge pass is not operational success. The judge sees a reference answer and truncated trajectory text, and the cross-family sample is not an unseen test set.

The comparison is conditional on historical runs. A full-codebase audit found call/result association, direct/MCP description parity, effective decoding-setting, resumed-process, mutable trial-state, verifier-fallback, and empty-plan issues. Prefix caching and bundled serving settings further confound component attribution. We do not infer that every historical row encountered every defect. Subsequent source repairs do not establish which historical trials were affected or retroactively validate their outputs; scoped impact assessment and revalidation remain required before causal or deployment conclusions.

An offline paired re-scoring of 2,419 stored trajectory identities changes judge pass labels on 573 pairs (76.3% agreement). This sensitivity analysis provides neither human ground truth nor a repaired-actor or all-61 result; the full paired comparison is reported in Appendix B.

The 31 scenarios are authored rather than a probability sample of maintenance work. The later 61-scenario expansion was not evaluated, and the 70B rows do not provide balanced cross-family replication. Accordingly, the benchmark supports conditional systems evaluation, not maintenance approval or equipment control.

5 Conclusion

SmartGridBench provides repeatable synthetic fixtures for separating interface, planning, latency, and judged-quality questions. The recorded Core31 and six-scenario results are useful only with their cohort denominators and validity caveats. Future work must repair and revalidate the harness, regenerate the failure taxonomy, obtain independent human/domain labels, and evaluate any larger scenario expansion under a frozen manifest before making broader claims.

AI-use disclosure

AI coding assistants (Claude and Codex) were used throughout this work: to write and debug the benchmark, harness, and server code; to design and review the experiment workflow; to write the analysis scripts and run evidence and provenance checks over the recorded artifacts; to draft and revise this manuscript, including the LaTeX markup of every table; to check literature and citations; and to help articulate the measurement limitations reported in the Validity and Limitations section. Numeric table contents are transcribed from the outputs of the analysis scripts and were re-verified against those outputs; no reported value was produced by a language model asserting it.

Generative models also proposed candidate scenarios through the project's LiteLLM-based generation workflow. All five generated scenarios retained in the frozen 36-file resource were produced by `watsonx/meta-llama/llama-3-3-70b-instruct`: three from prompt version `v0.1` and two from `v0.2`. Each was retained only after a per-scenario rubric review that rejected fabricated ground truth, repaired the intended tool sequence, and re-selected the hand-authored comparator; the applied edits are recorded per scenario in the resource. Ten further generated candidates were

rejected. The remaining 31 scenarios of the frozen resource, and the entire 31-scenario quality core, are hand-authored.

The agent configurations under test and the LLM judges that score them are research instruments, described with the task and configurations and treated as measurements throughout; they are not part of this disclosure of authoring assistance. AI assistance, automated checks, and model-based review do not constitute independent human or domain-expert validation, and the paper makes no such claim. The authors reviewed the AI-assisted code, analyses, tables, citations, and text, checked reported values against the recorded experiment outputs, and take full responsibility for the content of this paper. No AI system is an author.

REFERENCES

Anthropic. Introducing the model context protocol. https://www.anthropic.com/news/model-context-protocol, 2024. Open protocol for LLM tool integration.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Meta. Llama-3.1-8B-Instruct model card, 2024a. URL https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct.

Meta. Llama-3.3-70B-Instruct model card, 2024b. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

Meta. Llama-4-Maverick-17B-128E-Instruct model card, 2025. URL https://huggingface.co/meta-llama/Llama-4-Maverick-17B-128E-Instruct.

Dhaval Patel, Shuxin Lin, James Rayfield, Nianjun Zhou, Roman Vaculin, Natalia Martinez, Fearghal O'Donncha, and Jayant Kalagnanam. AssetOpsBench: Benchmarking ai agents for task automation in industrial asset operations and maintenance, 2025. URL https://arxiv.org/abs/2506.03828v1.

A NEGATIVE SCHEMA FIXTURES AND AUDIT SUPPLEMENTS

The frozen resource contains five negative schema fixtures used only to test rejection of invalid inputs. They are not maintenance tasks and are excluded from the denominators for 36 validated files, the 31-scenario quality core, and the later 61-scenario repository state.

A provisional recomputation of the audit-derived taxonomy, applying the hallucination dimension's occurrence polarity to the stored failure rows, contains 1,966 historical failure rows, 1,276 paper-eligible, with paper-eligible labels 944 task completion, 182 retrieval, 78 sequence, and 72 hallucinations. These counts come from that recomputation over the stored score rows, not from a published export: the canonical taxonomy export still carries the uncorrected polarity, under which the fourth label reads as result verification. The prior manual-audit strata have not been revalidated against the relabeling. No taxonomy prevalence or corrected-figure claim is made. It is not a new all-61 result or causal mitigation evidence. The 15-scenario intervention matrix and per-dimension judge diagnostics remain subject to review and revalidation.

B HISTORICAL JUDGE SENSITIVITY

We reparsed saved responses from the legacy rubric v1 with DeepSeek Flash (reasoning off) and compared them with stored WatsonX Maverick judgments on identical trajectory identities. The analysis contains 2,419 paired identities (2,391 newly scored and 28 reused) out of 2,420 registry-admitted identities; one mitigation plan–execute identity was disk-ambiguous and excluded without retry. DeepSeek's mean score was 0.4305 with 768/2,419 passes, versus 0.5542 with 1,143/2,419 passes for the historical judge. Pass labels agreed on 1,846/2,419 pairs (76.3%); 573 pairs disagreed, with 474 historical passes becoming DeepSeek failures and 99 historical failures becoming DeepSeek passes. Exact stored scores agreed on 863/2,419 pairs.

Table 4: Historical judge sensitivity by actor arm. Each denominator is the paired subset for that row. Scores and passes are judge outputs, not operational success or human labels.

Actor arm	n	DS mean	Hist. mean	DS pass	Hist. pass	Disagree
Agent-as-Tool	675	0.4035	0.4778	177	275	138
Plan–Execute	834	0.4343	0.5813	276	407	195
Verified PE	910	0.4471	0.5861	315	461	240

Agreement is not accuracy: neither judge supplies human ground truth, and the legacy rubric omits hidden scenario ground truth. The 474 Agent-as-Tool multi-call histories retain output-association uncertainty; disagreement is 97/474 in those histories versus 41/201 in the other Agent-as-Tool histories. This does not prove swaps or estimate their incidence. Repeated scenarios and trials are not independent human-validation samples, so no significance, causal, repaired-actor, or all-61 claim follows. A separate thinking-high side experiment is protocol-feasibility evidence only (five reused scores, two prior missing responses, one new length failure, and 2,412 never-started requests), not a representative thinking-arm comparison or task score.

C REVISION ANALYSIS AND EVIDENCE BOUNDARIES

The private revision analysis records the SHA-256 digests of its inputs and reconstructs the historical core table from the evidence registry and per-trial score records. It checks the matched scenario membership before computing contrasts, uses a fixed bootstrap seed, and tests that comparing a condition with itself produces zero difference. The latency output retains per-scenario summaries so that pooled quantiles can be distinguished from task-specific variation.

The analysis scripts and frozen output tables are retained in the private revision record; an anonymous release package remains to be finalized. Re-running the analysis requires the source evidence files; changing those inputs creates a new analysis, not a byte-identical reproduction of this one. The new uncertainty estimates are a September revision analysis of existing trials, not newly collected model evaluations.

The exploratory September cross-family judge pilot evaluates saved trajectories using capped and full inputs. Its frozen requests, raw responses, settings, source hashes, and token usage are retained with the private analysis. Recomputing its summaries does not require new API calls; repeating inference may produce different judgments even with identical inputs. The 28 sampled trials contain 27 distinct trajectories and produce 30 distinct request payloads across the 56 calls. Two trials of the same scenario have identical trajectories; their four identical-payload calls yield one pass and three failures. Trial count therefore overstates distinct evidence in this sample.

The later check in Section 3.3 uses 21 cases held out from settings calibration, but previously seen in that exploratory pilot. Its headline comparison uses the first draw with historically capped inputs; it is distinct from the pilot's capped/full-input comparison and is not an unseen test set.

A separate annotation sample is prepared for independent outcome assessment. No new human labels or expert judgments are claimed. Model agreement and the existing failure-label audit do not substitute for those assessments.

D POST-HOC UNCERTAINTY ANALYSIS

For the revision, we resample the 31 matched scenarios with replacement and retain all five observed trials on both sides of each contrast. We compute 10,000 paired scenario-cluster bootstrap replicates with seed 20260917 and report percentile 95% intervals. Scenario-level pairing reflects that the same tasks occur in each configuration; trial indices are not assumed to represent matched random seeds. These are post-hoc, unadjusted intervals for four selected comparisons, not preregistered confirmatory tests.

Table 5: Post-hoc differences relative to the MCP Agent-as-Tool baseline (B). Pass-rate differences are in percentage points. Brackets contain percentile 95% scenario-cluster bootstrap intervals.

Comparison minus B	Mean-score difference	Pass-rate difference
Direct (A)	$0.032\ [-0.026, 0.089]$	$6.5\ [-3.2, 15.5]$
Optimized MCP (C)	$0.031\ [-0.010, 0.075]$	$2.6\ [-3.2, 9.0]$
Plan-Execute (Y)	$0.140\ [0.026, 0.256]$	$12.3\ [-7.1, 31.0]$
Verified PE (Z)	$0.157\ [0.045, 0.270]$	$18.7\ [0.0, 36.8]$

The direct-versus-MCP intervals include zero for both outcomes. The observed ten-pass difference therefore does not support a confident quality ranking. Planning has positive mean-score intervals relative to the MCP baseline, but the pass-rate intervals are much wider: Plan-Execute includes zero, and the Verified Plan-Execute lower endpoint is zero. We avoid treating the larger point estimates as precise improvements.

These intervals describe sensitivity to the composition of this small authored benchmark under scenario resampling. The scenarios were not a probability sample of maintenance work, so the intervals do not quantify performance uncertainty on real transformers. They also omit uncertainty about judge validity, dataset construction, and shared implementation errors. Resampling the observed trials cannot repair those sources of bias.