How Do Tool-Augmented LLM Agents Perform on Real-World Energy Analytics Tasks?

Anonymous Author(s)
Affiliation
Address
email

Abstract

While agentic benchmarks have emerged across both general-purpose and domain-specific settings including finance, coding, law, and drug discovery, energy-domain evaluations remain limited to static knowledge recall. This is a critical gap for a sector that demands live data retrieval, specialized regulatory and market knowledge, and multi-step quantitative reasoning under real-world constraints. Despite its complexity and societal importance, the energy sector remains substantially underserved relative to domains where dynamic, tool-augmented evaluation has matured considerably.

We present an empirical study of tool-augmented LLM agents on real-world energy market analytics tasks. Our evaluation environment consists of 212 expert-curated problems spanning three broad categories: (1) Market Data Retrieval and Analysis, (2) Knowledge Retrieval and Interpretation, and (3) Advanced Quantitative Modeling and Decision Analytics, encompassing tasks such as price and demand analysis, tariff impact modeling, asset revenue estimation, and optimization modeling, each graded across multiple difficulty levels.

Agents are provided with a configurable suite of domain tools including live electricity market APIs for major U.S. ISOs, regulatory docket search, utility tariff databases, asset optimization models, and retrieval-augmented generation over energy market documents. To assess the performance of the agents along multiple dimensions, we employ a multi-dimensional evaluation protocol that scores responses on approach correctness, answer accuracy, attribute alignment, and source validity, with category-aware routing to match scoring criteria to question type. We evaluate both closed-source and open-source LLMs, offering a comparative analysis of how model capability and domain tooling interact in a high-stakes professional domain, with all artifacts publicly released.

1 Introduction

The energy sector is one of the most analytically demanding domains for AI-assisted decision support. Energy market professionals routinely synthesize heterogeneous, time-sensitive information spanning real-time market prices, complex regulatory frameworks, asset-level financial models, and datasets from ISO/RTO market systems, utility tariffs, interconnection queues, and weather services. Analysts at utilities, independent power producers, regulators, and consulting firms execute these workflows under conditions where errors carry material financial and operational consequences. Large language models (LLMs) have demonstrated strong capabilities in natural language understanding, knowledge retrieval, and structured reasoning [1, 2, 3]. The emergence of tool-augmented agentic frameworks—where models iteratively invoke external tools to retrieve data and execute computations—has further expanded the potential of LLMs in professional analytical workflows [4, 5]. Domain-specific

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

benchmarks have begun evaluating such systems in finance, law, software engineering, and drug discovery [6, 7, 8, 9, 10].

Despite this progress, the energy sector remains largely absent from rigorous agentic evaluation. Most prior AI work in energy focuses on predictive tasks such as load forecasting, renewable generation estimation, and electricity price prediction using supervised learning on historical data [11, 12]. WattWorks, a benchmark from the Electric Power Research Institute (EPRI), evaluates LLMs on power system questions but does not assess tool-augmented agents performing multi-step analytical workflows reflective of real analyst practice [13]. Consequently, no benchmark currently evaluates whether LLM agents can execute end-to-end energy analytics workflows under realistic operational constraints. To address this gap, we introduce EnergyEvals, an evaluation framework for tool-augmented LLM agents on real-world energy analytics tasks. The first iteration focuses on U.S. electricity market analytics, with future versions expanding to additional regions and energy sub-domains. This paper makes the following contributions:

- **A domain benchmark of 212 expert-curated tasks** spanning three core capability areas – market data retrieval and analysis, knowledge retrieval and interpretation, and advanced quantitative modeling and decision analytics. Each of these categories contains tasks across three difficulty levels (Easy, Medium, and Hard), which were generated by practitioners with doctoral-level training and combined industry experience exceeding 25 years at leading energy consulting and engineering organizations.
- **A configurable agentic evaluation environment** providing agents with nine domain-specific tools, including live ISO/RTO market APIs covering all major U.S. wholesale markets, utility tariff databases, regulatory docket search, renewable energy generation simulation, battery revenue optimization, and retrieval-augmented generation over electricity market reports and market protocol documents.
- **A multi-dimensional evaluation protocol** with category-aware rubric routing that assesses approach correctness, answer accuracy, attribute alignment, and source validity through an LLM-as-a-judge framework calibrated to specific quality requirements defined by energy analytics domain experts.
- **An empirical study of seven frontier LLMs** spanning closed-source and open-source models, revealing model-specific performance profiles and failure modes that emerge exclusively under realistic agentic task execution in a high-stakes professional domain.
- **Public release** of the benchmark dataset, evaluation framework, agent execution traces, and scoring code to support reproducibility and community extension.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 describes the benchmark dataset. Section 4 presents the agent architecture and tool suite. Section 5 defines the evaluation protocol. Section 6 reports experimental results. Conclusions and next steps are covered in Section 7.

2 Related Work

Research on benchmarks for agentic large language model (LLM) systems has expanded rapidly as tool-use frameworks mature. Early general-purpose benchmarks demonstrate that real-world multi-step reasoning remains difficult even for frontier models. GAIA evaluates web-augmented reasoning tasks where human non-experts solve 92% of problems while state-of-the-art models score below 30% [14], and SWE-bench measures software engineering agents solving real GitHub issues [9]. Benchmarks such as AgentBench, ToolBench, τ-bench, and TheAgentCompany consistently reveal large gaps between model reasoning ability and successful task completion across interactive environments, API ecosystems, and simulated enterprise settings [15, 16, 17, 18]. AstaBench further demonstrates that autonomous scientific discovery remains unsolved [19].

Domain-specific agentic benchmarks have emerged across professional fields, revealing that general capability improvements do not reliably transfer to specialized domains. In finance and enterprise analytics, Finance Agent Benchmark and InvestorBench show that even top

models achieve only moderate accuracy [7, 20], while CLASSIC and EnterpriseBench highlight agent struggles with workflow orchestration and tool usage [21, 22]. PaperBench and ScienceAgentBench demonstrate that replicating academic research remains extremely challenging [23, 24], and across productivity and specialized domains—including OdysseyBench, ContextBench, MedAgentBench, and LegalAgentBench—performance degrades significantly as tasks require deeper contextual understanding or domain expertise [25, 26, 27, 28]. SkillsBench covers energy through only three narrow power-system tasks, yet even with curated Skills agents fail more than half of them (47.5% pass rate) and domain knowledge gaps emerge as a primary failure mode, further motivating purpose-built evaluation frameworks that pair domain-specific tools with tasks representative of real energy analyst workflows [29].

Within the energy domain, most AI research has focused on predictive and optimization tasks such as load forecasting, electricity price forecasting, and renewable generation modeling [11, 12, 30]. Recent surveys highlight the absence of robust agentic evaluation frameworks for real-world analytical workflows [31]. WattWorks from EPRI shows that frontier LLMs perform well on multiple-choice power-sector questions (around 83–86% accuracy) but decline by roughly 27 percentage points on open-ended technical tasks [13], while ElecBench and smart-grid agent frameworks confirm that retrieval-augmented tools improve but do not fully resolve operational challenges [32, 33].

Tool-augmented language agents offer a promising paradigm for such workflows. ReAct demonstrated that interleaving reasoning with tool invocation enables iterative solution refinement [4], and Toolformer showed that models can learn autonomous tool invocation [5]. Sandboxed code execution and structured tool access further expand agent capabilities [34], though poor tool integration can introduce new reasoning errors [35]. Recent frameworks therefore emphasize multi-dimensional scoring and LLM-as-a-judge methodologies to better diagnose performance [36, 37, 38]. Building on this work, EnergyEvals evaluates tool-augmented agents on real-world energy analytics tasks involving live market data retrieval, regulatory analysis, and optimization modeling absent from existing benchmarks.

3 Benchmark Dataset

3.1 Design Philosophy

The dataset is designed to evaluate whether tool-augmented LLM agents can execute realistic, end-to-end energy analytics workflows in a professional domain where accuracy, traceability, and quantitative rigor are essential. Rather than testing benchmark-style factual recall, tasks mirror the workflows of practicing energy market analysts, including retrieving live pricing data, interpreting formal regulatory and interconnection documents, and executing multi-step financial models under operationally realistic constraints. Task development was led by domain experts with doctoral-level training and prior professional experience at organizations including McKinsey, ICF, LCG Consulting, and General Electric, representing more than 25 years of combined industry experience. This practitioner grounding is reflected in the prompt design through the use of market-specific terminology (e.g., nodal pricing, ancillary service qualification thresholds, interconnection milestones) and operational constraints such as efficiency parameters, degradation costs, state-of-charge limits, and IRR targets that are typical of real client engagements rather than academic exercises. The current release is intentionally scoped to U.S. electricity markets—covering both deregulated and vertically integrated systems—to ensure cross-task comparability while preserving real-world complexity arising from differences in market design, tariff structures, and regulatory and interconnection documentation, with future releases planned to expand coverage to additional geographies and adjacent energy sub-domains.

3.2 Capability Areas

The 212-task corpus is organized around three broad capability areas across three difficulty levels (Easy, Medium, Hard), with 107 Data, 86 Knowledge, and 19 Quant. tasks respectively (see Appendix A.1 for the full breakdown).

1. **Market Data Retrieval and Analysis ("Data").** Tasks requiring extraction, aggregation, filtering, and formatting of structured market data from ISO/RTO databases and APIs. Representative analyst functions include day-ahead and real-time price analysis, ancillary service performance evaluation, load and generation dispatch reporting, and

cross-market comparisons. Example: *"Show me the monthly average of day-ahead prices for ERCOT Houston hub in 2023 based on your ERCOT database."*

2. **Knowledge Retrieval and Interpretation ("Knowledge").** Tasks requiring navigation of formal regulatory documents, utility tariff filings, market operation manuals, and interconnection procedures to answer precise procedural and structural questions. These tasks test the agent's ability to identify authoritative sources, locate relevant provisions, and interpret regulatory language accurately without fabricating content. Example: *"What are the fees associated with each milestone in the ERCOT generation interconnection process based on the ERCOT fee schedule and Resource Interconnection Handbook?"*

3. **Advanced Quantitative Modeling and Decision Analytics ("Quant").** Tasks requiring multi-step analytical reasoning, modeling, and optimization under explicit operational assumptions. Representative functions include battery energy storage revenue estimation, demand-charge impact assessment, internal rate of return (IRR) computation, and optimization-based decision support with explicit constraint specifications. Example: *"If a 4-hour battery earns revenues from arbitrage only in ERCOT West hub over 15 years, what should the $/MW capex be to earn a 13% IRR? Assume 81% roundtrip efficiency, $25/MWh degradation cost, state-of-charge limits of 10–90%, and use prices from 2010–2024 as the representative 15-year window."*

3.3 Difficulty Stratification

Tasks are stratified across three difficulty levels to probe increasingly demanding agent behaviors:

- **Easy** - Direct retrieval tasks with explicit source context and limited data transformation. The agent must select and invoke the correct tool but requires minimal multi-step reasoning. Example: *"What detailed fees are associated with each decision point in the NYISO generation interconnection process based on NYISO Manuals 23 and UG21?"*
- **Medium** - Tasks requiring retrieval with or without explicit source hints, combined with moderate aggregation, filtering, or cross-attribute comparison. Example: *"Which PJM price hub had the highest day-ahead average price in January 2024 based on your PJM database?"*
- **Hard** - Tasks requiring multi-step, multi-source reasoning and advanced quantitative modeling under realistic operational assumptions. Agents must correctly sequence tool calls, apply domain-specific constraints, and integrate outputs across multiple reasoning steps. Example: *"If a 4-hour battery earns revenues from arbitrage only in ERCOT West hub over 15 years, what should the $/MW capex be to earn a 13% IRR? Assume 81% roundtrip efficiency, $25/MWh degradation cost, state-of-charge limits of 10–90%, and use prices from 2010–2024 as the representative 15-year window."*

3.4 Paired Prompt Construction

A central design feature of the dataset is *paired prompt construction*: selected tasks are available in two variants – one explicitly specifying the information source, and one omitting source specification. This enables controlled evaluation of source-scaffolding effects on agent performance under matched semantic intent. For example, *"What are the participation requirements for regulation service in CAISO based on the latest Business Practice Manual for Market Operations?"* is a task with a specified source. *"What are the participation requirements for regulation service in CAISO?"* is the without-source counterpart.

4 Agent Architecture and Tool Suite

4.1 ReAct Agent Framework

Agents are implemented as ReAct-style reasoning-and-acting agents that execute an iterative Thought → Action → Observation loop [4]. At each step, the agent produces a natural language reasoning trace (*Thought*), selects and invokes a tool with structured arguments (*Action*), and receives the tool's structured output (*Observation*). The loop terminates when the agent produces a final answer or reaches a configurable maximum iteration budget. This architecture is expressive enough to represent multi-hop retrieval chains, sequential

computation pipelines, and iterative refinement strategies without constraining the agent to an execution pattern (see Appendix A.2 for a conceptual view of the architecture and Appendix A.4 for implementation details).

4.2 Model Configurations

Seven frontier LLMs are evaluated as agent backends (see Appendix A.3 for the full configuration table). Closed-source models (GPT-5.2, GPT-5-mini, Gemini-3.1-Pro, Claude Sonnet 4.6) and open-source models (Kimi-K2.5, Qwen3-Max-Thinking, DeepSeek-V3.2) are all configured with low reasoning effort and temperature 0 for deterministic, reproducible outputs, using off-the-shelf inference APIs without domain adaptation.

4.3 Tool Suite

Agents are given access to a suite of domain-specific tools grouped under nine categories spanning live structured market data (GridStatus API, Database MCP), formal document retrieval (RAG MCP, Dockets, Web Search), domain computation (Battery Optimization, Renewables), and contextual supplementary data (Tariffs, Weather). Tools are registered in a typed registry and exposed as structured JSON Schema function definitions, enabling identical execution across all models. A full tool inventory and descriptions are provided in Appendix A.7.

All agent executions are traced at the step level as structured JSON artifacts (see Appendix A.6), enabling the failure mode analyses in Section 6 and released as a secondary research artifact. Output traces and raw evaluation reports are included in the Github repository (https://anonymous.4open.science/r/energyevals-3F76/)

5 Evaluation Protocol

5.1 Evaluation Dimensions

Agent responses are assessed across three complementary dimensions that together capture the distinct quality requirements that are typical in the energy analytics domain. Each dimension targets a failure mode that matters in practice but would be invisible under aggregate accuracy-only scoring.

1. **Approach Correctness (1–5).** Does the agent employ an appropriate analytical strategy? This dimension evaluates tool selection, sequencing logic, and whether the agent's reasoning pathway is consistent with how a professional analyst would approach the task. An agent that reaches a correct numerical answer through an inappropriate pathway (for example, by hallucinating data rather than retrieving it from the correct API) receives reduced credit on this dimension.

2. **Answer Accuracy / Attribute Alignment (0–1).** Is the final answer factually correct, and does it satisfy all specified constraints, including temporal scope, geographic jurisdiction, entity type, and units of measurement? For tasks that are only quantitative in nature, accuracy is considered based on the difference between the ground truth and the agent's response within an acceptable absolute or relative tolerance. For tasks with a combination of quantitative and qualitative components, a set of up to 5 expected attributes (specific numerical values, named entities, or conclusions) are extracted from the ground truth with an LLM judge and manually reviewed and updated as needed by domain experts. The same LLM judge is then used to extract attributes from the agent's answer and compare each of the extracted attributes with the expected attributes to while considering the defined absolute (e.g. $\epsilon_{\text{abs}} = \pm 2$) or relative tolerances (e.g. $\epsilon_{\text{rel}} = \pm 10\,\%$) for numerical attributes. The score equals matched / total attributes, yielding a continuous value in $[0, 1]$. This dimension captures failures where an agent retrieves valid data but for the wrong ISO, wrong time period, or wrong entity, as well as simple factual errors.

3. **Source Validity (1–5).** Are the data sources cited or implicitly relied upon real, appropriate, and accessible? This dimension penalizes hallucinated source citations, fabricated document version numbers, use of inappropriate or outdated sources, and failures to ground answers in tool-retrieved evidence where the task requires it.

5.2 Category-Aware Rubric Routing

Rubric emphasis is adapted to each capability area, reflecting the differential importance of evaluation dimensions across task types. For *Data Retrieval and Analysis* and *Advanced Quantitative Modeling* tasks, Answer Accuracy is important as a measure of the agent's correctness, since the outcomes of the agent's analysis are expected to closely match the ground truth if all goes well. However, for *Knowledge Retrieval and Interpretation* tasks, Attribute Alignment provides a better measure of correctness given that the ground truth will contain multiple attributes that a solid response from the agent is expected to match. Source Validity and Approach Correctness are critical for all the task categories as it is important to verify that the agent is arriving at the answers in a logical way and not via lucky hallucinations.

Rubrics are applied using a GPT-5-mini judge with access to the ground truth answer, the full agent execution trace, and a structured scoring rubric (see Appendix A.8). The LLM-as-a-judge approach is well-suited to the open-ended, multi-part responses characteristic of professional analytics tasks, which resist reduction to exact-match or template-based scoring [36]. Judge outputs include a numeric score on each dimension and a natural language justification, enabling qualitative audit of scoring decisions.

5.3 Reported Metrics

The main reported metrics are as follows.

- **Overall class-balanced means for each dimension:** $\bar{a}_m$, $\bar{c}_m$, $\bar{v}_m$ aggregated across the 212 tasks considering each category and difficulty level combination. For each model m and question q, the judge produces scores on three dimensions: Approach Correctness $a_{m,q} \in [1, 5]$, Answer Accuracy $c_{m,q} \in [0, 1]$, and Source Validity $v_{m,q} \in [1, 5]$. We report class-balanced scores (i.e., weighted average score per category with equal weighting applied to each category) per dimension, independently without aggregation into a composite score. Each dimension captures a distinct failure mode and collapsing them into a single number would obscure the performance profiles that are important to observe. Also, the class-balanced scores account for the different total number of questions in each category. See Appendix A.9, A.10, and, A.11 for a breakdown by capability areas.
- **Efficiency and cost metrics:** Includes total tokens per question, tool calls, and cost per question. These are reported as simple averages. The cost is estimated using input, output, and cached tokens without changes to the default caching behavior of the models. Latency is excluded because network round-trip times vary per provider API and are not a model-capability metric.
- **Failure rate:** Reflects the percentage of tasks that failed based on three failure mode definitions - maxed out iterations, context window limitations, clarification requests. This is also reported as a class-balanced metric to avoid placing too much weight on categories with more but easier questions.

No confidence intervals are reported because the benchmark uses a single trial per question. Within-question variance is zero, and cross-question standard error is not a meaningful uncertainty estimate.

6 Results and Analysis

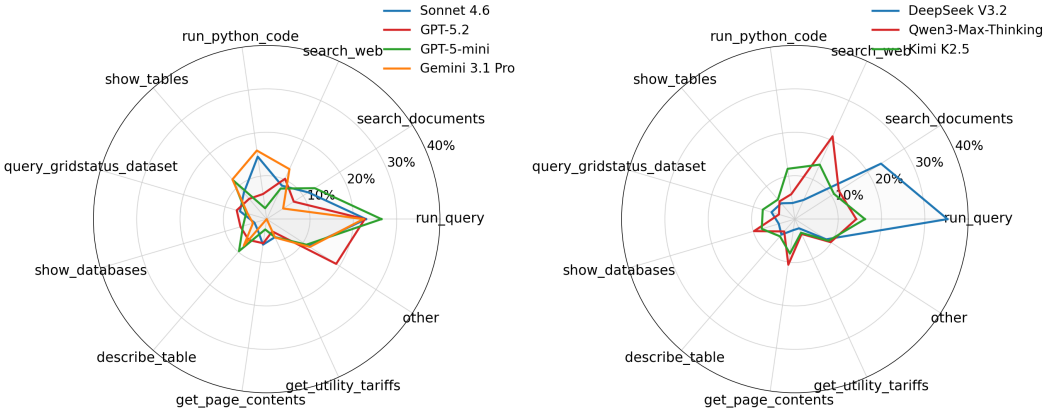
6.1 Overall Results

Table 1 provides a summary of the class-balanced evaluation metrics across all capability areas and difficulty stratifications for the seven frontier models considered. A breakdown of the class-balanced metrics for each capability area is included in the Appendix section (see Appendix A.9, A.10, and A.11). Figure 1 shows the distribution of tool usage across the different models. The key takeaways from Table 1 and Figure 1 are as follows.

1. **Closed-Source Models Lead, but Overall Performance Remains Unsaturated.** Considering the accuracy results (range: 0 to 1) in Table 1, GPT-5.2, Gemini-3.1-Pro, and Claude Sonnet 4.6 have the best performance values—62%, 61%, and 58%, respectively. The best-performing open-source model is Kimi-K2.5 with a score of 51%, which is 7 percentage points lower than Claude Sonnet 4.6. However, Kimi-K2.5 achieved this

Table 1: Evaluation Metrics Across Models

Model	Accuracy	Approach	Source Validity	Tokens	Tool Calls	Cost Estimate ($)	Failure Rate (%)
GPT-5.2	0.62	4.13	3.45	223k	8.31	0.14	2.92
GPT-5-mini	0.41	3.65	2.75	106k	3.72	0.01	18.42
Gemini-3.1-Pro	0.61	3.98	2.36	292k	5.8	0.19	3.83
Claude Sonnet 4.6	0.58	4.06	2.20	254k	6.91	0.79	5.88
Kimi-K2.5	0.51	4.04	2.23	375k	9.71	0.06	15.87
Qwen3-max-thinking	0.39	3.70	2.34	291k	7.16	0.36	1.92
DeepSeek V3.2	0.44	3.95	2.13	486k	12.33	0.08	22.83



(a) Closed source models

(b) Open source models

Figure 1: Tool use distribution for Closed Source vs Open Source models

performance at 42%, 30%, and 7% of the costs of GPT-5.2, Gemini-3.1-Pro, and Claude Sonnet 4.6, respectively. The best-performing model, based on accuracy, still has a 38% improvement margin, suggesting that domain expertise will continue to play an important role in designing agentic systems with high accuracy guarantees for energy-domain applications. It is worth noting that Qwen3-Max-Thinking is a cost outlier among open-source models due to its unified reasoning architecture, which defaults to extended chain-of-thought generation and thus incurs substantially higher token costs than standard open-source models.

2. **Models Exhibit a Planning–Execution Gap in Agentic Tasks.** The Approach results (range: 0 to 5) from Table 1 show that both closed-source and open-source models generally perform reasonably well. GPT-5-mini, which has the lowest performance, has a score equivalent to 73% (i.e., 3.65/5) compared to a maximum accuracy of 62%. These results further corroborate the observation that expert guidance remains important in designing agentic systems for energy-domain applications. While both open and closed models are capable of proposing reasonable analytical approaches, reliable execution requires domain-specific knowledge to structure workflows, ensure correct dataset usage, and define appropriate validation criteria.

3. **Source Attribution Is Not an Emergent Behavior in Current Models.** The Source Validity scores (range: 0 to 5) in Table 1 are generally low across both closed-source and open-source models. These low scores arise because the models do not always include clear source links by default, which could hinder reproducibility of outcomes and limit trust—both of which are particularly important in this context. Explicit prompting with expert guidance will be required to achieve the desired source referencing behavior.

4. **Impact of Context Window Size on Task Completion and Accuracy.** The models with the highest failure rates are DeepSeek-V3.2, GPT-5-mini and Kimi-K2.5, as shown in Table 1. The context windows for DeepSeek-V3.2 and Kimi-K2.5 are 163k and 262k tokens, respectively, compared to at least 400k tokens for GPT-5.2, Gemini 3.1 Pro, and Claude

Sonnet 4.6 models [39, 40, 41, 42]. These shorter context windows could be a limitation, especially for knowledge retrieval tasks that require processing significant amounts of information and multi-step problems that require longer contexts to preserve information from each step. Although the Qwen3-Max-Thinking model also has a relatively shorter context window (256k tokens), it interestingly has a considerably smaller failure rate (i.e., 1.92%). However, its overall accuracy score is significantly lower (i.e., 0.39), implying that tasks that did not fail based on the three failure mode definitions (i.e., maxed-out iterations, context window limitations, clarification requests) still produced low-accuracy outcomes. GPT-5-mini's failure mode is largely due to excessive requests for additional clarification, which violates the instructions in the system prompt (see Appendix A.8).

5. **Higher Token Usage Is Not Correlated with Improved Performance.** From Table 1, the best-performing model also had the lowest average token usage and corresponding lowest cost among comparable models, supporting the claim that higher token usage does not necessarily translate to better performance.

6. **Tool Selection Bias Varies Between Open and Closed-Source Models.** The tool usage distribution charts (Figure 1) show "run_query" as the dominant tool across the models. This is because multiple tasks require retrieval of data from the energy markets database. Excluding the "run_query" tool, closed-source models show a more balanced internal tool mix, while open-source models are relatively more retrieval-heavy (e.g., "search_web", "search_documents").

6.2 Performance with and without sources specified

As highlighted in the benchmark dataset design philosophy section, paired prompt construction is employed. This provides the basis for measuring the impact of explicitly including sources or tools to use in each question. Table 2 shows the class-balanced accuracy and source validity metrics for a subset of 61 tasks (see Appendix A.13 for the task IDs) that have clear counterparts with and without source. The table presents an interesting observation—the inclusion of sources does not always translate into improved accuracy across all models. This is due to a combination of the models' reasoning capabilities and the nature of the questions and tools provided, making it more likely for the models to take similar steps when answering the questions with or without sources specified. Figure 2 corroborates this, as the distribution of tool usage for both with- and without-source questions is practically the same. A step-by-step view of the traces across the seven models for a pair of questions (see Appendix A.14) also confirms this. However, the performance of source validity increases for questions with sources, as expected.

Table 2: Evaluation Results With and Without Sources

Model	**Without Sources**		**With Sources**	
	Accuracy	**Source Validity**	**Accuracy**	**Source Validity**
GPT-5.2	0.70	3.16	0.71	4.04
GPT-5-mini	0.57	2.44	0.53	2.95
Gemini-3.1-Pro	0.70	2.13	0.66	2.77
Claude Sonnet 4.6	0.73	2.04	0.75	2.49
Kimi-K2.5	0.64	2.06	0.60	2.55
Qwen3-Max-Thinking	0.50	2.40	0.49	2.65
DeepSeek-V3.2	0.59	1.94	0.62	2.44

6.3 Performance with and without selected domain-specific tools

A subset of 30 tasks (see Appendix A.13 for the task IDs) was selected from the overall dataset to evaluate performance without domain-specific tools. The 30 tasks are the most challenging in the dataset and are at Medium and Hard difficulty levels across the three capability areas. The Accuracy and Approach metrics are shown in Table 3. The results clearly show that agents perform better when given access to the domain tools across all the models considered. In some cases, the accuracy scores double, emphasizing the importance of domain tools in agentic applications for the energy domain.

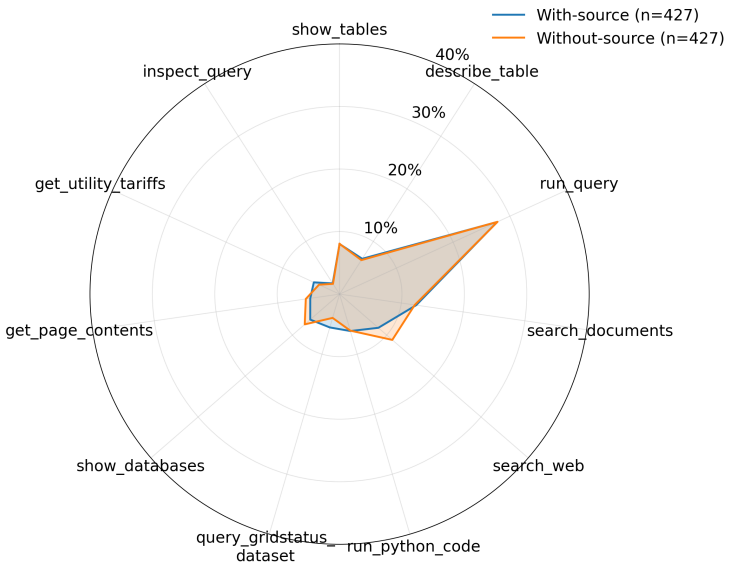


Figure 2: Tool use distribution without sources vs with sources specified (n represents 61 questions across 7 models)

Table 3: Evaluation Results With and Without Domain-Specific Tools

Model	Without Tools		With Tools	
	Accuracy	Approach	Accuracy	Approach
GPT-5.2	0.36	3.44	0.55	3.72
GPT-5-mini	0.07	2.90	0.25	3.83
Gemini-3.1-Pro	0.42	3.16	0.51	3.74
Claude Sonnet 4.6	0.32	3.47	0.68	4.01
Kimi-K2.5	0.17	3.47	0.30	3.33
Qwen3-Max-Thinking	0.10	3.08	0.34	3.37
DeepSeek-V3.2	0.16	2.93	0.24	3.59

7 Conclusion

In previous sections, a rich discussion regarding the performance of tool-augmented LLM agents on real-world energy analytics tasks was presented. The insights show that while tool-augmented agents can tackle some real-world energy analytics tasks, expert guidance is still required to improve the quality of outcomes produced by those agents. This domain expert guidance is what is required to take the performance of these agents from generally good to exceptionally insightful.

We also noted that there are some limitations associated with this first iteration of EnergyEvals and, as such, the following will be captured in future iterations.

1. **Regional and sub-domain expansion.** The dataset considered in this first iteration of EnergyEvals focuses on the US and tasks relating to electricity markets. While this represents a good starting point, energy analytics is a global phenomenon and goes beyond electricity-related tasks. Subsequent iterations of EnergyEvals will include tasks covering new regions and other energy sub-domains.

2. **Performance under high reasoning model configurations.** The models considered in this iteration have low reasoning configurations to evaluate performance under resource-constrained scenarios. However, it is possible that other reasoning configuration levels can improve performance. We will investigate this in subsequent iterations.

References

[1] OpenAI. GPT-4 technical report. Technical report, OpenAI, 2024. URL https:

//arxiv.org/abs/2303.08774.

[2] Anthropic. The Claude 3 model family: Opus, Sonnet, Haiku. Technical report, Anthropic, 2024. URL https://api.semanticscholar.org/CorpusID:268232499.

[3] Google DeepMind. Gemini: A family of highly capable multimodal models. Technical report, Google DeepMind, 2024. URL https://arxiv.org/abs/2312.11805.

[4] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.03629.

[5] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2302.04761.

[6] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models, 2025. URL https://arxiv.org/abs/2306.06031.

[7] Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. Finance agent benchmark: Benchmarking llms on real-world financial research tasks, 2025. URL https://arxiv.org/abs/2508.00828.

[8] Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2308.11462.

[9] Carlos E. Jiménez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.06770.

[10] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. ChemCrow: Augmenting large language models with chemistry tools, 2023. URL https://arxiv.org/abs/2304.05376.

[11] Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016. doi: https://doi.org/10.1016/j.ijforecast.2015.11.011.

[12] Rafał Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4):1030–1081, 2014. doi: https://doi.org/10.1016/j.ijforecast.2014.08.008.

[13] Electric Power Research Institute (EPRI). Benchmarking large language models for the electric power sector. White Paper 3002034347, Electric Power Research Institute, Palo Alto, CA, 2025.

[14] Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2311.12983.

[15] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianhao Shen, Yuxiao Dong, Jie Tang, and Yann LeCun. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023. URL https://arxiv.org/abs/2308.03688.

[16] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2307.16789.

[17] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024. URL https://arxiv.org/abs/2406.12045.

[18] Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2412.14161.

[19] Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi, Sergey Feldman, Dany Haddad, Jena D. Hwang, Peter Jansen, Varsha Kishore, Bodhisattwa Prasad Majumder, Aakanksha Naik, Sigal Rahamimov, Kyle Richardson, Amanpreet Singh, Harshit Surana, Aryeh Tiktinsky, Rosni Vasu, Guy Wiener, Chloe Anastasiades, Stefan Candra, Jason Dunkelberger, Dan Emery, Rob Evans, Malachi Hamada, Regan Huff, Rodney Kinney, Matt Latzke, Jaron Lochner, Ruben Lozano-Aguilera, Cecile Nguyen, Smita Rao, Amber Tanaka, Brooke Vlahos, Peter Clark, Doug Downey, Yoav Goldberg, Ashish Sabharwal, and Daniel S. Weld. AstaBench: Rigorous benchmarking of AI agents with a scientific research suite, 2024. URL https://arxiv.org/abs/2510.21652.

[20] Haohang Li, Yupeng Cao, Yangyang Yu, Shashidhar Reddy Javaji, Zhiyang Deng, Yueru He, Yuechen Jiang, Zining Zhu, Koduvayur Subbalakshmi, Guojun Xiong, Jimin Huang, Lingfei Qian, Xueqing Peng, Qianqian Xie, and Jordan W. Suchow. InvestorBench: A benchmark for financial decision-making tasks with LLM-based agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, Vienna, Austria, 2025. URL https://aclanthology.org/2025.acl-long.126.

[21] Michael Wornow, Vaishnav Garodia, and Vasilis Vassalos. Top of the CLASS: Benchmarking LLM agents on real-world enterprise tasks. In *ICLR 2025 Workshop on Building Trust in LLMs and LLM Applications*, 2025. URL https://openreview.net/forum?id=RQjUpeINII.

[22] Harsh Vishwakarma, Ankush Agarwal, Ojas Patil, Chaitanya Devaguptapu, and Mahesh Chandran. Can LLMs help you at work? A sandbox for evaluating LLM agents in enterprise environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. URL https://arxiv.org/abs/2510.27287.

[23] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research, 2025. URL https://arxiv.org/abs/2504.01848.

[24] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin

Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.05080.

[25] Weixuan Wang, Dongge Han, Daniel Madrigal Díaz, Jin Xu, Victor Rühle, and Saravan Rajmohan. OdysseyBench: Evaluating LLM agents on long-horizon complex office application workflows, 2025. URL https://arxiv.org/abs/2508.09124.

[26] Han Li, Letian Zhu, Bohan Zhang, Rili Feng, Jiaming Wang, Yue Pan, Earl T. Barr, Federica Sarro, Zhaoyang Chu, and He Ye. ContextBench: A benchmark for context retrieval in coding agents, 2026. URL https://arxiv.org/abs/2602.05892.

[27] Yixing Jiang, Kameron C. Black, Gloria Geng, Danny Park, James Zou, Andrew Y. Ng, and Jonathan H. Chen. MedAgentBench: A realistic virtual EHR environment to benchmark medical LLM agents. *NEJM AI*, 2025. URL https://arxiv.org/abs/2501.14654.

[28] Haitao Li, Junjie Chen, Jingli Yang, Qingyao Ai, Wei Jia, Youfeng Liu, Kai Lin, Yueyue Wu, Guozhi Yuan, Yiran Hu, Wuyue Wang, Yiqun Liu, and Minlie Huang. LegalAgentBench: Evaluating LLM agents in legal domain. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pages 2322–2344, Vienna, Austria, 2025. URL https://aclanthology.org/2025.acl-long.116.

[29] Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, Shuyi Wang, Binxu Li, Qunhong Zeng, Di Wang, Xuandong Zhao, Yuanli Wang, Roey Ben Chaim, Zonglin Di, Yipeng Gao, Junwei He, Yizhuo He, Liqiang Jing, Luyang Kong, Xin Lan, Jiachen Li, Songlin Li, Yijiang Li, Yueqian Lin, Xinyi Liu, Xuanqing Liu, Haoran Lyu, Ze Ma, Bowei Wang, Runhui Wang, Tianyu Wang, Wengao Ye, Yue Zhang, Hanwen Xing, Yiqi Xue, Steven Dillmann, and Han-chung Lee. Skillsbench: Benchmarking how well agent skills work across diverse tasks, 2026.

[30] Stefan Pfenninger and Iain Staffell. Long-term patterns of european PV output using 30 years of validated hourly reanalysis and satellite data. *Energy*, 114:1251–1265, 2016. doi: https://doi.org/10.1016/j.energy.2016.08.060.

[31] Furqan Amjad, Tarmo Korõtko, and Argo Rosin. Review of llms applications in electrical power and energy systems. *IEEE Access*, 13:150951–150969, 2025. doi: 10.1109/ACCESS.2025.3599922.

[32] Xiyuan Zhou, Huan Zhao, Yuheng Cheng, Yuji Cao, Gaoqi Liang, Guolong Liu, Wenxuan Liu, Yan Xu, and Junhua Zhao. ElecBench: A power dispatch evaluation benchmark for large language models. *arXiv preprint arXiv:2407.05365*, 2024. URL https://arxiv.org/abs/2407.05365.

[33] Sai Santhosh Polagani. AI agents for smart grid operations and renewable energy management. *Iconic Research and Engineering Journals*, 8(11):1278–1292, 2025. URL https://www.irejournals.com/formatedpaper/1708600.pdf.

[34] Daixuan Cheng, Shaohan Huang, Yuxian Gu, Huatong Song, Guoxin Chen, Li Dong, Wayne Xin Zhao, Ji-Rong Wen, and Furu Wei. LLM-in-Sandbox elicits general agentic intelligence, 2026. URL https://arxiv.org/abs/2601.16206.

[35] Botao Yu, Frazier N. Baker, Ziru Chen, Garrett Herb, Boyu Gou, Daniel Adu-Ampratwum, Xia Ning, and Huan Sun. Tooling or not tooling? the impact of tools on language agents for chemistry problem solving. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. URL https://arxiv.org/abs/2411.07228.

[36] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2306.05685.

[37] Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023. URL https://arxiv.org/abs/2305.14387.

[38] Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. Benchmarking foundation models with language-model-as-an-examiner. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc. URL https://doi.org/10.48550/arXiv.2306.04181.

[39] DeepInfra. Deepinfra models library, 2026. URL https://deepinfra.com/models. Accessed: 2026-03-24.

[40] OpenAI. Models. https://developers.openai.com/api/docs/models, 2026. OpenAI API Documentation. Accessed: 2026-03-24.

[41] Google DeepMind. Gemini pro, 2026. URL https://deepmind.google/models/gemini/pro/. Model page describing the Gemini Pro family of multimodal AI models. Accessed: 2026-03-24.

[42] Anthropic. Claude sonnet, 2026. URL https://www.anthropic.com/claude/sonnet. Anthropic model page describing the Claude Sonnet family of models. Accessed: 2026-03-24.

A Appendices

A.1 Dataset Breakdown

Table A.1: Dataset breakdown by capability area and difficulty level (n=212)

Capability Area	Easy	Medium	Hard	Total
Data	13	61	33	107
Knowledge	40	43	3	86
Quant.	0	2	17	19

A.2 Conceptual view of overall evaluation pipeline

An illustration of the overall evaluation pipeline is as shown below.

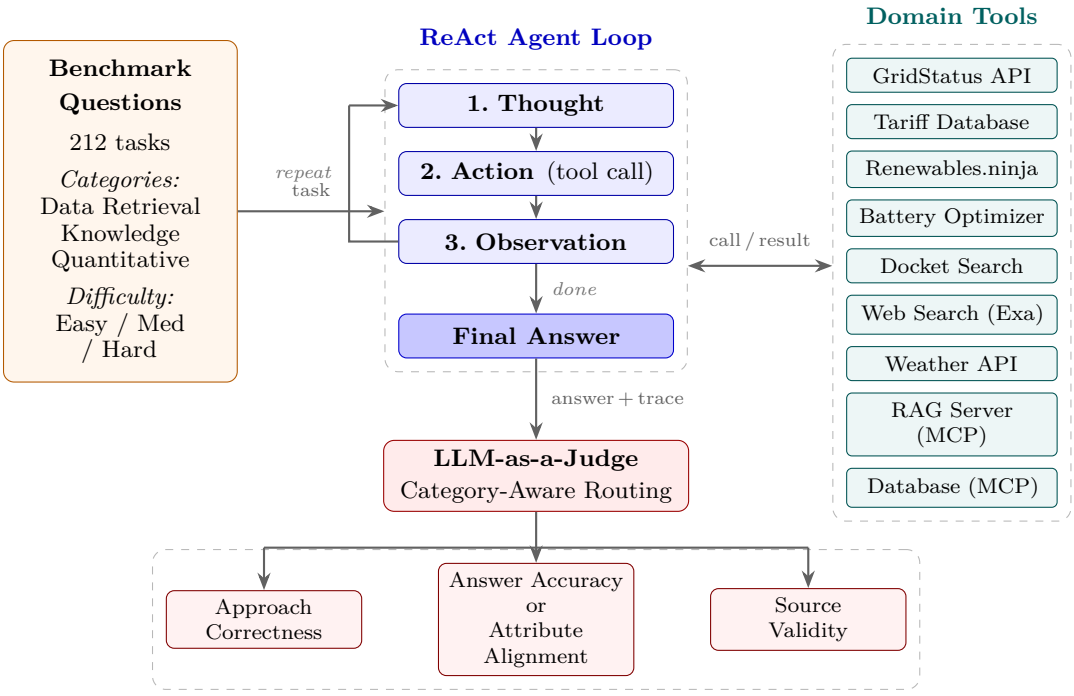


Figure A.1: Conceptual overview of the evaluation pipeline. A question from the benchmark dataset is presented to a ReAct agent backed by one of seven LLMs. The agent iterates through **Thought**, **Action** (tool call), and **Observation** steps, invoking domain tools as needed, until it emits a Final Answer. The answer and execution trace are then scored by an LLM-as-a-Judge across four dimensions using category-aware rubric routing.

A.3 Model Configurations

Table A.2: Models evaluated and inference configurations

Model	Provider	Open Source	Reasoning	Reasoning Level
GPT-5.2	OpenAI	No	Yes	Low
GPT-5-mini	OpenAI	No	Yes	Low
Gemini-3.1-Pro	Google	No	Yes	Low
Claude Sonnet 4.6	Anthropic	No	Yes	Low
Kimi-K2.5	Moonshot	Yes	Yes	N/A
Qwen3-Max-Thinking	Alibaba	Yes	Yes	N/A
DeepSeek-V3.2	DeepSeek	Yes	Yes	N/A

GPT, Gemini, and Sonnet models are configured with low reasoning effort to evaluate performance under compute-efficient inference conditions. In subsequent releases, high reasoning level configurations will be examined. All models are evaluated with temperature set as 0 for deterministic, reproducible outputs. No system-level fine-tuning or domain adaptation is applied; all models are used off-the-shelf via their respective inference APIs (with DeepInfra APIs used for all the open source models [39]).

A.4 Agent Implementation Details

The ReAct agent is implemented through a provider abstraction layer that wraps model-specific API differences—including function-calling schemas, tool-use message blocks, response parsing logic, and streaming behavior—behind a common interface. This design enables identical benchmark execution across all evaluated models without modification to the agent loop or tool suite.

A.5 Tool Suite Overview

Table A.3: Tool suite available to agents

Tool Category	Data Source	Coverage
GridStatus API	GridStatus.io	All US wholesale electricity markets
Tariffs	OpenEI Tariffs API	U.S. utility tariffs
Renewables	Renewables.ninja	Solar/wind generation simulation
Battery Optim.	N/A	Arbitrage-only revenues for battery projects
Dockets	FERC; state PUCs	Federal and 7 other state jurisdictions
Web Search	Exa API	Open web
Weather	OpenWeatherMap	Current & forecast
RAG (MCP)	Document corpus	Market reports and manuals
Database (MCP)	Market data portals	ERCOT, NYISO, PJM and ISONE markets

A.6 Observability and Trace Collection

All agent executions are traced at the step level, capturing the complete sequence of thought, action, observation, and answer events as structured JSON artifacts. Traces record tool call arguments and raw responses alongside token-level timing and iteration counts. These traces serve two functions: they enable the failure mode discussions in Section 6, and they constitute a secondary research artifact released alongside the benchmark.

A.7 Tool description

Table A.4: Tool inventory grouped by category

Tool category	Tool	Description
System	list_files	Lists files/directories in a specified path, optionally recursively.
	grep_files	Searches files for text patterns with optional glob/ path filters.
	run_python_code	Executes Python code in a sandboxed environment and returns output/errors.
	run_shell_command	Executes shell commands in a controlled environment and returns stdout/stderr.
GridStatus API	list_gridstatus_datasets	Lists available GridStatus datasets (ID, name, description).
	inspect_gridstatus_dataset	Returns schema/metadata for a specific GridStatus dataset.
	query_gridstatus_dataset	Queries a GridStatus dataset with filters/-time bounds and returns results.
Tariffs	get_utility_tariffs	Retrieves utility tariff/rate records from OpenEI IURDB.
Renewables	get_solar_profile	Returns hourly solar generation profile (capacity factors) for a location/date range.
	get_wind_profile	Returns hourly wind generation profile (capacity factors) for a location/date range.
Battery Optim.	battery_revenue_optimization	Solves battery dispatch/arbitrage optimization and outputs revenue metrics/profile.
Dockets	search_ferc_dockets	Searches FERC dockets/filings.
	search_dc_dockets	Searches District of Columbia PSC dockets.
	search_maryland_dockets	Searches Maryland PSC dockets.
	search_new_york_dockets	Searches New York PSC dockets.
	search_north_carolina_dockets	Searches North Carolina Utilities Commission dockets.
	search_south_carolina_dockets	Searches South Carolina PSC dockets.
	search_texas_dockets	Searches Texas PUCT dockets/filings.
	search_virginia_dockets	Searches Virginia SCC dockets.
Web Search	search_web	Runs web search over external sources.
	get_page_contents	Fetches and extracts content from specified URLs.
Weather	geocode_location	Converts location names to latitude/longitude.
	get_current_weather	Returns current weather conditions for a location.
	get_forecast	Returns short-term weather forecast for a location.
	get_historical_weather	Returns historical weather over a specified period.
	get_air_pollution	Returns air-quality and pollutant metrics for a location.
RAG (MCP)	search_documents	Retrieves relevant passages from indexed document corpora (MCP RAG).
Database (MCP)	show_databases	Lists accessible databases.
	show_tables	Lists tables in a selected database/schema.
	describe_table	Returns table schema/column metadata.
	show_indexes	Shows table indexes for query planning.
	run_query	Executes SQL query against connected database.
	inspect_query	Provides query inspection/validation metadata.
	preview_table	Returns a row preview/sample from a table.

A.8 System and evaluation prompts

Here are the different prompts used for the benchmark and evaluation runs. They are also included in the publicly available repository.

Agent System Prompt

```
You are an Expert Energy Analyst.
Use your best effort to answer each question with only one attempt.
No room for back and forths with the user
```

Judge System Prompt

```
You are a strict evaluator of answers relating to energy markets
analysis.
Follow expert industry standards.
Your output MUST exactly match the provided output schema.
Do not add extra fields or surrounding text.
```

Approach Evaluation Prompt

```
You are evaluating the approach correctness of how an AI agent
obtained answers to an energy market related question and not the
correctness of the answer itself.
In addition to question, you also have a summary of the suggested
approach provided by an expert and a trace of the steps the agent
took to answer the question which you can use to infer the agent's
approach to answering the question
Question:
{question}
Suggested Approach (Ground Truth):
{suggested_steps}
Agent's Steps:
{agent_steps_trace}
Evaluate:
     • Correct problem framing
     • Appropriate data sources (ISO postings, tariffs, settlement
       data, APIs)
     • Logical analytical steps
     • Correct tool usage (if applicable)
Rating scale:
5=expert-like, 4=minor issues, 3=notable gaps, 2=major flaws,
1=wrong approach
```

Accuracy Evaluation Prompt

```
You are evaluating the factual and numerical accuracy of an AI
agent's answer to a question relating to energy markets analysis.
Question:
{question}
Expected Answer (Ground Truth):
{expected_answer}
Agent's Answer:
{agent_answer}
```

Evaluate:
- Numerical correctness (values, sign, magnitude, units, time basis)
- Factual alignment (market/ISO, node/zone, product, settlement type etc.)
- Completeness of key facts

Tolerance:
Allow $\leq$ {abs_tol} absolute error OR $\leq$ {rel_tol}% relative error unless exactness is required.

Source Evaluation Prompt

You are evaluating the following two things only.
1. Explicit inclusion of sources in an AI agent's answer to a question relating to energy markets analysis.
2. Relevance of the included sources for the question
You can extract or infer relevant sources from the question itself or from the suggested approach ground truth
Do not penalize for not explicitly adding queries or code for pulling data for verification as long as the source specified is consistent with what the agent has access to and is plausible
Internal databases are based on data from authoritative external sources and as such, the internal databases are equivalent to external authoritative sources (e.g. market portals) and should be treated as such
Question:
{question}
Suggested Steps:
{suggested_steps}
Agent's Answer:
{agent_answer}
Evaluate:
- Authority of sources
- Alignment with expected sources
- Appropriateness for the claim
- Missing citations when required

Attribue Evaluation Prompt

You are evaluating attribute alignment of an AI agent's answer against a canonical set of expected attributes.
Question:
{question}
Expected Attributes (canonical, JSON):
{expected_attributes_json}
Agent's Answer:
{agent_answer}
For each expected attribute, decide whether the agent answer contains the correct value or a reasonable equivalent, respecting units and time basis.
Tolerance:
For numeric attributes, allow $\leq$ {abs_tol} absolute error OR $\leq$ {rel_tol}% relative error unless exactness is required.

Attribute Extraction Prompt

```
You are generating a canonical attribute set for evaluating an AI
agent answer to an energy market question.
Extract no more than 5 high-value attributes from the expected
answer.
Each attribute should be specific, evaluable, and tied to the
question intent.
Prefer attributes that are most critical to correctness.
Question:
{question}
Expected Answer:
{expected_answer}
```

A.9 Results for market data retrieval and analysis tasks

Table A.5: Evaluation Metrics for Market Data Retrieval and Analysis Tasks

Model	Accuracy	Approach	Source Validity	Tokens	Tool Calls	Cost Estimate ($)	Failure Rate (%)
GPT-5.2	0.71	4.32	3.62	275k	10.28	0.14	0
GPT-5-mini	0.53	4.02	2.69	108k	4.58	0.01	11.34
Gemini-3.1-Pro	0.71	4.01	2.44	273k	6.58	0.18	1.23
Claude Sonnet 4.6	0.65	4.05	2.10	261k	7.18	0.80	1.23
Kimi-K2.5	0.60	4.15	2.29	464k	11.19	0.06	8.10
Qwen3-Max-Thinking	0.42	3.76	2.59	315k	8.4	0.38	0.62
DeepSeek V3.2	0.56	4.10	2.24	560k	14.93	0.09	6.79

Compared to the values in Table 1, the accuracy scores for this category are higher implying that the models generally perform better on this category of questions. Also failure rates are generally lower showing that the models were able to successfully complete more tasks in this category. The outlier is GPT-5-mini whose failure mode is significantly related to clarification requests and that is category-independent.

A.10 Results for knowledge retrieval and interpretation tasks

Table A.6: Evaluation Metrics for Knowledge Retrieval and Interpretation Tasks

Model	Accuracy	Approach	Source Validity	Tokens	Tool Calls	Cost Estimate ($)	Failure Rate (%)
GPT-5.2	0.52	4.05	3.78	150k	4.41	0.13	1.16
GPT-5-mini	0.33	3.87	2.73	102k	2.05	0.02	12.80
Gemini-3.1-Pro	0.50	4.00	2.40	259k	3.6	0.17	2.33
Claude Sonnet 4.6	0.59	4.22	2.77	203k	4.13	0.64	0
Kimi-K2.5	0.48	4.33	2.58	246k	5.47	0.05	10.47
Qwen3-Max-Thinking	0.46	4.17	2.52	226k	3.92	0.28	1.16
DeepSeek V3.2	0.49	3.98	2.45	370k	7.07	0.07	9.30

Scores in Table A2 are generally lower than the overall numbers in Table 1 implying lower performance for this category.

A.11 Results for advanced quantitative modeling and decision analytics tasks

As expected, Table A3 shows lower accuracy scores, higher token usage, and significantly higher failure rates, showing that the questions under this category are more challenging compared to the other two categories.

A.12 Public framework and data repository overview

The public repository (available here - https://anonymous.4open.science/r/energyevals-3F76/) contains a complete list of the 212 questions. However, benchmark traces, ground truths, evaluation results and justifications are released for a subset

Table A.7: Evaluation Metrics for Advanced Quant Modeling and Decision Analytics Tasks

Model	Accuracy	Approach	Source Validity	Tokens	Tool Calls	Cost Estimate ($)	Failure Rate (%)
GPT-5.2	0.54	3.84	2.79	257k	14.89	0.17	10.52
GPT-5-mini	0.26	2.68	2.89	111k	6.47	0.01	10.53
Gemini-3.1-Pro	0.61	3.98	2.36	292k	5.8	0.19	3.83
Claude Sonnet 4.6	0.45	3.89	1.84	443k	18	1.43	21.05
Kimi-K2.5	0.36	3.52	1.74	457k	20.63	0.07	36.84
Qwen3-max-thinking	0.26	3.11	1.68	452k	14.79	0.56	5.26
DeepSeek V3.2	0.16	3.63	1.58	601k	21.58	0.1	68.42

containing 30 questions. This is to prevent data contamination and model overfitting issues. As subsequent versions of EnergyEvals become available, the datasets and codes will be updated.

A.13 Task IDs for source and tools impact analysis

The task IDs for the 61 tasks without sources considered for the source impact analysis are as follows. The ids follow the format: *With Source (Without Source Counterpart).*

With and Without source task IDs

```
1 (22), 2 (23), 3 (24), 6 (27), 15 (36), 16 (37), 17 (38), 18 (39),
19 (40), 20 (41), 21 (42), 67 (45), 68 (46), 69 (47), 70 (48), 71
(49), 72 (50), 73 (51), 83 (61), 84 (62), 85 (63), 86 (64), 87 (101),
88 (102), 89 (103), 90 (104), 91 (105), 92 (106), 93 (107), 95 (109),
96 (110), 97 (111), 98 (112), 100 (114), 123 (152), 124 (153), 125
(154), 126 (155), 127 (156), 128 (157), 129 (158), 130 (159), 131
(160), 132 (161), 133 (162), 134 (163), 144 (172), 149 (177), 150
(178), 151 (179), 188 (213), 189 (214), 191 (216), 196 (221), 197
(222), 198 (223), 200 (225), 201 (226), 202 (227), 203 (228), 206
(231)
```

The task IDs for the 30 tasks considered for the tool impact analysis are as follows.

Tool impact analysis task IDs

```
111, 112, 115, 172, 207, 209, 213, 214, 215, 216, 217, 218, 219, 220,
221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234,
235, 245
```

A.14 Sample trace path of a pair of questions with and without sources

The trace paths for the answers to questions 88 (with source) and 102 (without source variant) across all seven models evaluated are shown in the figure below.

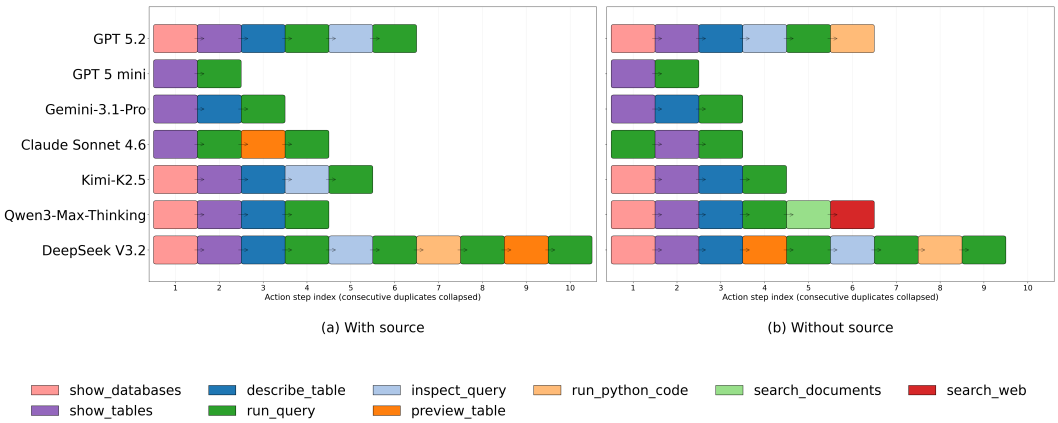


Figure A.2: Trace paths for all 7 models to answer the question (a) *"What was the difference between average ERCOT weekday and weekend day-ahead prices in the summer of 2023 based on your ERCOT database?"* and (b) *"What was the difference between average ERCOT weekday and weekend day-ahead prices in the summer of 2023?*