Pacific Northwest
NATIONAL LABORATORY

PNNL-39202

GridCoPilot for Thermal Events: An LLM-Based Platform for Power Grid Reliability Analysis

May 2026

1 Sarthak Chaturvedi
2 Kyung-bin Kwon
3 Shrirang Abhyankar
4 Palak Mattoo
5 Travis Thurber
6 Heng Wan
7 Casey Burleyson
8 Nathalie Voisin

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

U.S. DEPARTMENT of ENERGY

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-39202

GridCoPilot for Thermal Events: An LLM-Based Platform for Power Grid Reliability Analysis

May 2026

1 Sarthak Chaturvedi
2 Kyung-bin Kwon
3 Shrirang Abhyankar
4 Palak Mattoo
5 Travis Thurber
6 Heng Wan
7 Casey Burleyson
8 Nathalie Voisin

Prepared for
the U.S. Department of Energy
under Contract DE□AC05□76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Large Language Models show promise for translating natural language into database queries, but deploying such systems in safety-critical domains requires high reliability. We present an application of GridCoPilot to thermal event analysis (heatwaves and coldwaves) that affect power grid reliability. Our approach uses a LangChain SQL Agent to translate natural language queries into auditable SQL statements, with deterministic visualization routines that parse the structured query results. We introduce structural framing as a design principle, we integrate a NERC-region-level event library with county-level meteorology and decompose the combined data into three relational tables (event metadata, county-level event details, and a county-to-NERC subregion mapping), using prompt-guided joins to direct the model toward correct multi-table queries. For two core analytical patterns (identifying worst events by region and by region-year), the system achieved 100 % SQL accuracy across all 16 NERC subregions and both event types (64 queries total). These results validate the approach for target use cases, though performance on diverse natural language formulations requires further investigation. We discuss design trade-offs, failure modes including JSON output truncation, and pathways for extending this approach to other hazard domains.

Summary

This report presents GridCoPilot for thermal event analysis, an LLM-based platform that supports power grid reliability studies by enabling natural language querying of heatwave and coldwave event data. The system integrates a NERC-region-level thermal events library with county-level meteorological data, decomposed into three focused relational tables to facilitate reliable SQL generation by a LangChain SQL Agent powered by GPT-4.1. Deterministic Plotly-based visualization routines render county-level choropleth maps and regional dashboard widgets without LLM involvement, ensuring reproducibility. Evaluation across 64 canonical benchmark queries demonstrated 100% SQL generation accuracy and 98.4% end-to-end accuracy, with the single failure attributed to JSON output truncation rather than incorrect query generation. The report documents the system architecture, data engineering strategy, failure mode analysis, and design trade-offs, and outlines pathways for extending this approach to additional hazard domains such as hydropower drought.

Acknowledgments

This work was supported by the U.S. Department of Energy. The authors gratefully acknowledge the contributions of the Pacific Northwest National Laboratory team.

Acronyms and Abbreviations

AWS	Amazon Web Services
FERC	Federal Energy Regulatory Commission
FIPS	Federal Information Processing Standards
JSON	JavaScript Object Notation
LLM	Large Language Model
NERC	North American Electric Reliability Corporation
NL2SQL	Natural Language to SQL
NL2VIS	Natural Language to Visualization
PNNL	Pacific Northwest National Laboratory
RDS	Relational Database Service
SME	Subject Matter Expert
SQL	Structured Query Language
TOC	Table of Contents

Contents

Figures

Tables

1.0 Introduction

The rapid advancement of Large Language Models (LLMs) has changed how domain experts interact with complex data systems. Recent systems have achieved success in translating natural language queries into structured database queries (NL2SQL) (Rajkumar, Li, and Bahdanau 2022; Liu et al. 2023) and generating data visualizations from textual descriptions (NL2VIS) (Luo et al. 2022; Narechania, Srinivasan, and Stasko 2021). These capabilities aim to make data analysis accessible by lowering the technical barrier for non-programmers, with the goal of enabling domain experts to interrogate data without requiring SQL or visualization programming expertise. However, deploying such systems in mission-critical domains where incorrect answers can have serious operational consequences remains challenging.Natural language interfaces to databases have been extensively studied, with modern approaches leveraging neural sequence-to-sequence models and, more recently, LLMs fine-tuned or prompted for SQL generation (Scholak, Schucher, and Bahdanau 2021; Pourreza and Rafiei 2023). HydroLLM (Kizilkaya et al. 2025), for example, followed the established pipeline of identifying data sources, developing the datasets with adequate categorization and tagging with the help of subject matter experts, training the model, evaluating the model, and evaluating the overall tool. HydroLLM was developed with over 8000 questions, exposing the technical challenges of selecting models based on the data size and type of expected answers. While these systems show promise in research-purpose settings, computational resources and scalability remain challenging for general and operational user-friendly applications. For decision-support applications, verifiability, reproducibility, and auditability are also necessary. Similarly, natural language to visualization systems like NL4DV (Narechania, Srinivasan, and Stasko 2021) and Lida (Dibia 2023) attempt to automate the generation of charts and dashboards from textual intent. These developments suggest an opportunity for domain-specific LLM systems that target a focused set of analytical questions while providing integrated visualization capabilities.A particularly pressing need exists for specialized LLM tools to support the exploration of thermal events in power grid reliability studies. Heatwaves and cold snaps typically cause peak load increases from air cooling and heating appliances, creating regional resource adequacy stress that utilities must anticipate (Panteli and Mancarella 2015; Ke et al. 2016). These extreme temperature events can also cause power outages through distribution system failures (Guddanti et al. 2025). Emerging reliability standards (FERC Orders 1920 and 896) require utilities to demonstrate preparedness for such events.Wan et al. (Wan et al. 2025) developed comprehensive datasets of heatwave and cold snap events across the continental United States, examining over 12 event definitions from the scientific literature. Their analysis found that event definitions were less impactful than the choice of climate data source and temperature metrics. Building on this work, we further categorized and tagged these datasets to support a focused set of questions that utilities may ask when selecting thermal events for reliability planning. These questions require synthesizing data across multiple dimensions: geographic granularity (county-level to regional), temporal scope (single events to multi-year trends), and severity metrics (temperature extremes, spatial coverage, duration). The underlying data is complex, comprising millions of granular time-series records linked to event metadata and spatial hierarchies, making direct SQL querying impractical for non-technical users. While LLMs offer a promising interface, naive applications suffer from hallucinations, irreproducibility, and lack of provenance, which are serious limitations for operational decision-making.

Contributions. This paper extends GridCoPilot (Chaturvedi et al. 2025) to thermal event analysis, demonstrating the systematic application of LLM-based data exploration in support of grid reliability studies. Our contributions are threefold. First, we provide *domain-specific data engineering* for thermal events, combining two source datasets at different spatial scales (a NERC-region-level event library (Wan, Burleyson, and Voisin 2025) and county-level meteorology (Burleyson, Thurber, and Vernon 2023)) and separating them into three relational tables (event metadata, county-level event details, and a FIPS-to-NERC subregion mapping) with prompt-guided joins that direct the SQL Agent toward correct multi-table queries across all 16 NERC subregions. Second, we present a rigorous *evaluation methodology* with 64 canonical benchmark queries that isolate SQL generation accuracy from downstream failures, revealing that the primary challenge lies in JSON output truncation rather than query generation. Third, we document *failure mode analysis* including JSON truncation patterns and geometry payload constraints, providing practical guidance for deploying LLM-based systems in geospatial domains. Deterministic visualization pipelines (Plotly with rule-based rendering) eliminate an entire class of LLM failure modes, narrowing reliability concerns to the SQL generation and structured output phases.

This work addresses the following technical challenges: (1) What SQL generation accuracy can a constrained NL2SQL system achieve for thermal event queries when using domain-specific data engineering? (2) What failure modes emerge in LLM-based structured data retrieval, and how can they be mitigated? (3) What accuracy and performance trade-offs arise from polygon simplification in county-level event maps? (4) What metadata and governance artifacts are required for LLM systems to answer definitional questions reliably?

This work extends GridCoPilot to thermal event analysis with several domain-specific adaptations: (1) integration of a NERC-region-level thermal events library (Wan, Burleyson, and Voisin 2025) with county-level meteorology (Burleyson, Thurber, and Vernon 2023), separated into three relational tables (event metadata, county-level event details, and a FIPS-to-NERC subregion mapping linking 3,000+ US counties to 16 regional planning areas); (2) deterministic Plotly-based visualization that renders county-level choropleth maps from structured JSON output; and (3) prompt engineering informed by observed failure modes, including explicit join instructions and anti-truncation directives.

Section 2 describes the system methodology, including architecture, data transformation strategy, and dual-query approach. Section 3 presents evaluation results, failure mode analysis, and verification of the foundational data retrieval layer. Section 4 discusses implications, design trade-offs, and pathways for generalization to other hazard domains. Section 5 outlines current limitations. Section 6 provides code and data availability information, and Section 7 concludes with future directions.

2.0 Methodology

GridCoPilot uses a modular architecture that converts natural language queries into verifiable data insights and visualizations. We prioritize reliability and auditability through careful data engineering and deterministic visualization pipelines. We first describe the system architecture, then the data flow and transformation, and finally the dual-strategy approach for query handling and visualization

2.1 System Architecture

GridCoPilot employs a linear, auditable pipeline that converts an operator's query into an executable SQL statement, retrieves structured data, and renders a corresponding visualization. This workflow, illustrated in Figure 2, is initiated through a web-based interface built with Streamlit. The prompt template structure is shown in Figure 1.

The core of the system is a LangChain SQL Agent powered by GPT-4.1 (deployed via Azure OpenAI). Our implementation and all reported results are specific to GPT-4.1; performance characteristics and failure modes may differ for other LLM providers or model versions. The agent receives the natural-language query along with a structured prompt containing database schema information, explicit join instructions, and output format requirements. The agent internally performs SQL generation, executes the query against a PostgreSQL database (deployed on AWS RDS), and returns the results as structured JSON. Critically, the prompt instructs the agent to return complete JSON records with all FIPS codes and associated data; the "never truncate FIPS records" instruction is essential for ensuring data completeness. We selected GPT-4.1 empirically for its larger context window, which reduces the risk of output truncation when queries return thousands of county-level records; however, broader context-window side effects such as context degradation and instruction-following drift were not systematically evaluated.

Visualization is handled by deterministic Python routines rather than LLM-generated code. Once the SQL Agent returns structured JSON results, a dedicated visualization module (visualization_county_v1.py) parses the JSON and generates Plotly choropleth maps. This module uses Shapely for geometry processing and implements Douglas-Peucker polygon simplification to manage payload sizes. The visualization logic is rule-based: given the query type and returned data structure, the appropriate map type and styling are selected programmatically. This design ensures reproducibility (the same query results always produce the same visualization) and eliminates a class of potential LLM failures.

Every answer can be traced back to an inspectable SQL statement, and the deterministic visualization pipeline makes presentation errors reproducible and debuggable. Domain experts can inspect the generated SQL before proceeding, and data engineers can optimize or manually adjust queries when needed.

SYSTEM PROMPT *LangChain SQL Agent (GPT-4.1)*

Context. You are analyzing thermal events from the weather events database. Translate natural language questions into SQL queries that retrieve event data with complete geographic coverage.

Schema. Five tables are available:
Metadata: `heat_wave_metadata`, `cold_wave_metadata`
Details: `heat_event_details`, `cold_event_details`
Mapping: `county_nerc_mapping` (FIPS → NERC subregion)

NERC Subregions. 16 regions: 1:AZ-NM-SNV, 3:ERCOT, 5:NEW ENGLAND, 9:DELTA, 12:VACAR, 17:RFC, ...

Output Format. Return structured JSON:
`{"data": [{"NERC_ID": ..., "start_date": ..., "records": [{"id": FIPS, "T": temp}, ...]}]}`

Critical: Never truncate FIPS records. All county records must be included in full.

User query: `{question}`

Figure 1. Prompt template structure for the LangChain SQL Agent, showing the key components: task context, database schema, NERC subregion mappings, output format specification, and the critical instruction preventing JSON truncation.

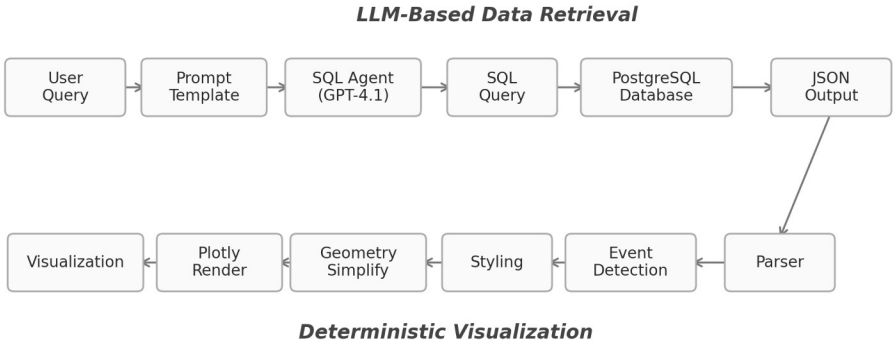

Figure 2. System architecture showing the separation between LLM-based data retrieval (top) and deterministic visualization (bottom). The SQL Agent generates queries against a PostgreSQL database, returning structured JSON that is processed by rule-based visualization routines without LLM involvement.

2.2 Data Flow and Transformation

Applying LLMs to domain-specific data raises a practical problem: the gap between how source data is organized and the SQL structures that LLMs can reliably produce. This work draws on two foundational datasets at different spatial scales: (1) the thermal events library compiled by Wan et al. (Wan, Burleyson, and Voisin 2025), which contains heatwave and coldsnap events identified using 12 literature-based definitions at the NERC subregion level, providing event-level metadata (event ID, type, peak temperature, spatial extent, duration, NERC subregion); and (2) the county-level hourly meteorology dataset from Burleyson et al. (Burleyson, Thurber, and Vernon 2023), which provides temperature observations at the county (FIPS code) level. For each event identified in the metadata, the county-level dataset supplies the granular temperature observations recorded at individual counties within the affected NERC subregion during the event period. Combining these two sources into a single flat table (as done in the source event library) yields a complete long-format representation, but exposing it directly to an

LLM-based SQL agent presents challenges: the combined volume of metadata, county-level details, and geographic mappings makes it difficult for the model to identify relevant columns and formulate targeted queries.

To make this data LLM-accessible, we decomposed and reorganized the combined source data into three relational table groups, as shown in Figure 3: (1) Event_Metadata tables (heat_wave_metadata, cold_wave_metadata) containing NERC-region-level event summaries derived from the thermal events library (Wan, Burleyson, and Voisin 2025); (2) Event_Details tables (heat_event_details, cold_event_details) containing county-level temperature records keyed by FIPS code and event ID, providing the finer-grained observations from the county-level meteorology data (Burleyson, Thurber, and Vernon 2023) for each identified event; and (3) a county_nerc_mapping table providing the explicit link from county FIPS codes to NERC subregion IDs. We use the 16 NERC subregion definitions from Wan et al. (Wan et al. 2025), which divide the continental US into planning areas that align with regional transmission organization boundaries. Note that while NERC has 6 regional entities, these 16 subregions provide finer geographic granularity for thermal event analysis.

This decomposition, shown in Figure 4, inverts the typical NL2SQL strategy of denormalizing data into flat tables. Instead of pre-joining everything to eliminate joins, we keep each table focused on a single analytical role and use the prompt template to spell out the join logic for the LLM. The prompt instructs the SQL Agent to: (a) query the appropriate metadata table to identify the target event, (b) retrieve county-level details using the event ID, and (c) join with county_nerc_mapping to ensure geographic consistency between the event's NERC region and the counties displayed. This has three practical benefits: (1) each table has a small, well-defined schema that the LLM can reason about without confusion; (2) the prompt-guided join pattern is predictable and aligns with few-shot examples; (3) the mapping table can be independently inspected and validated. We call this making the data LLM-accessible: optimized for reliable automated query generation rather than human database administration.

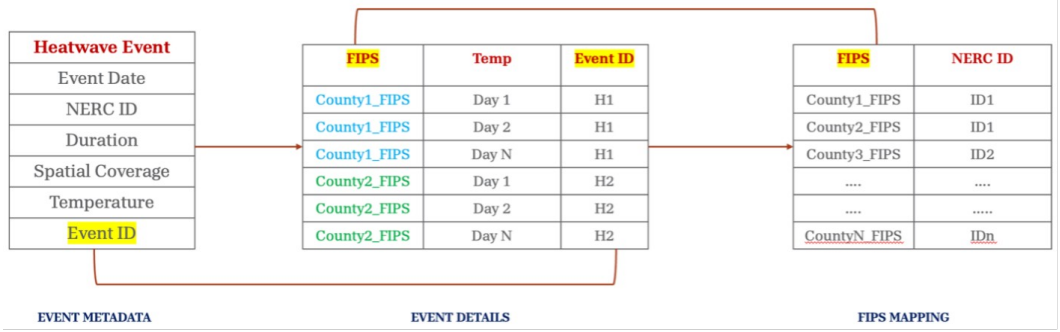

Figure 3. The normalization flow of the source data. Event Metadata is linked to Event Details via EventID, and Event Details are mapped to regional information through the FIPS Mapping table.

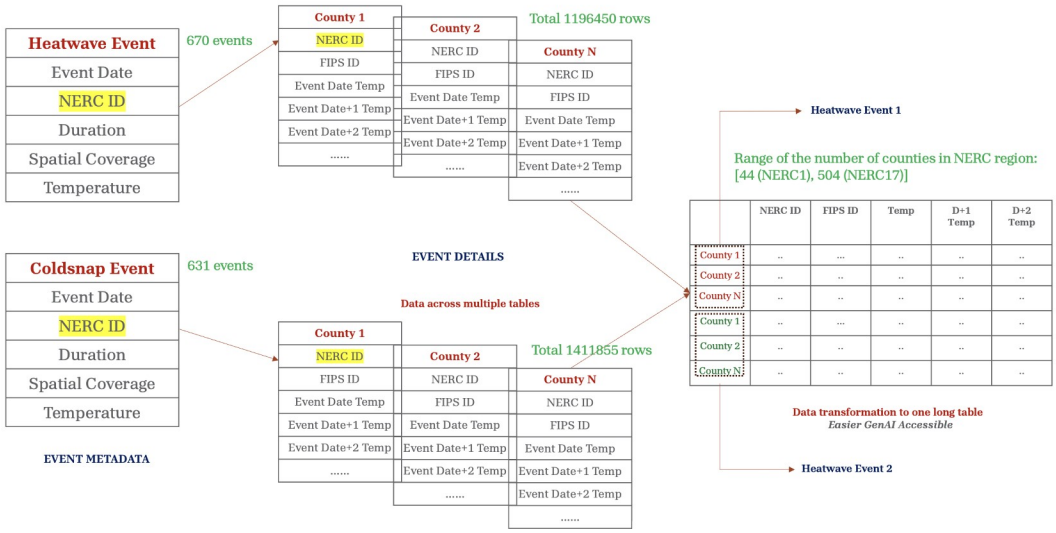

Figure 4. Data transformation: the source long-format data is separated into focused relational tables (metadata, details, mapping) to support LLM-accessible querying via prompt-guided joins.

2.3 A Dual-Strategy for Query and Visualization

Our architecture employs a dual-strategy approach to efficiently handle two distinct classes of user questions by leveraging different parts of the data structure. This bifurcation is informed by the observation that grid operators' information needs fall into two categories: detailed forensic analysis of specific events, and high-level situational awareness across multiple events.

Granular (county-level) questions are precise, data-dense asks that require detailed spatial resolution (e.g., “What was the worst coldsnap in NERC 10 in 1985?” or “Show me the daily temperature progression during event C42”). These queries use the event details and mapping tables through prompt-guided joins to perform deep analysis, typically returning thousands to tens of thousands of county-day observations. The standard output is a detailed, county-level choropleth map or animated time-series visualization, as shown in Figure 5.

However, rendering these high-resolution geospatial visualizations presents significant performance challenges. A single county polygon can contain hundreds or thousands of vertices, and rendering 3,000+ counties over multiple days can generate multi-megabyte JSON payloads that overwhelm browser memory and cause rendering failures. To ensure responsive performance, we implemented a pragmatic relaxation and approximation strategy using the Douglas-Peucker algorithm (Douglas and Peucker 1973) for polygon simplification. County boundary geometries are simplified using Shapely's simplify(tolerance=0.01, preserve_topology=True), which applies the Douglas-Peucker algorithm while maintaining topological validity. The tolerance parameter was selected empirically to balance visual recognizability with payload size for interactive browser rendering. Importantly, geometry simplification is applied only to the display layer; it does not alter event metadata, FIPS codes, temperature values, or county-to-NERC assignments returned by the SQL query. The analytical data pipeline and visualization geometry pipeline are fully decoupled. For animated visualizations spanning multiple days, Plotly animation frames are constructed in memory and

rendered with playback controls. A perfectly accurate map that fails to render provides no value, so trading some geometric precision for responsiveness is worthwhile in practice.

High-level (NERC-level) questions are aggregated queries focused on regional trends, rankings, and comparative analysis (e.g., “What are the five worst heatwave events in NERC 3?” or “Compare 2011 and 2012 summer event severity across all regions”). These questions do not require county-level granularity and can be resolved more efficiently by querying the compact Event_Metadata table directly. This lightweight approach bypasses the millions of granular records, enabling rapid generation of summary tables, bar charts, and regional dashboard widgets (Figure 6). Typical query response times are under 500ms, compared to 3–8 seconds for granular queries with map rendering. This allows operators to quickly gain situational awareness and identify events of interest before committing to a deeper, more computationally expensive granular drilldown.

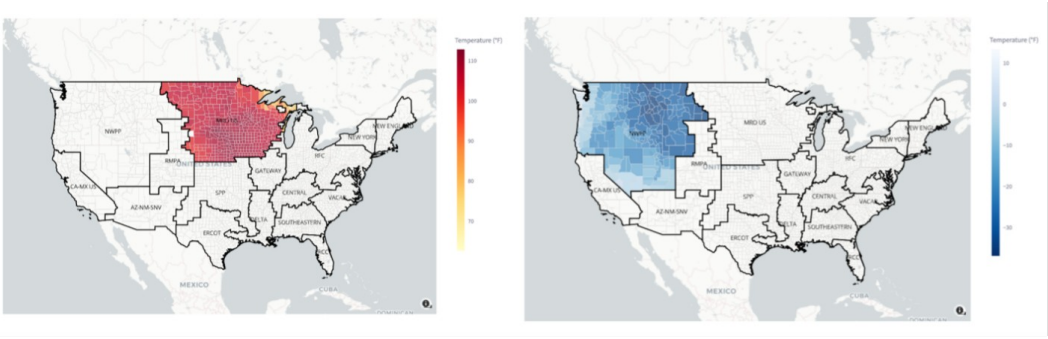

Figure 5. Example of a granular visualization showing county-level temperature extremes for a specific event. Rendering such high-fidelity maps necessitates memory optimization techniques.

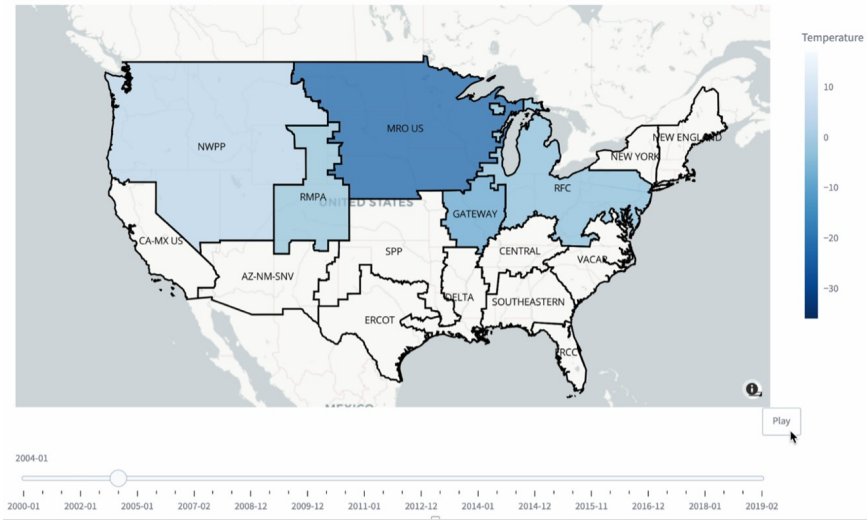

Figure 6. Example of a high-level visualization widget, showing top events by peak temperature and spatial coverage across NERC subregions, generated efficiently from metadata.

2.4 Structural Framing: Data Architecture Design Choices

The decomposed schema, precomputed geographic mapping, and prompt-guided join logic described above are all instances of what we call structural framing: constraining and organizing the problem space to match the LLM's strengths and avoid its weaknesses. Below we explain the rationale behind each choice.

Domain-Specific Schema Design. Rather than exposing the source long-format dataset directly to the LLM, we designed a decomposed schema optimized for the specific query patterns of thermal event analysis. The three-table structure separates concerns: metadata tables provide event-level summaries (event ID, type, severity metrics, duration), detail tables provide county-level temperature records keyed by FIPS and event ID, and the mapping table provides geographic context. Each table has a small, well-defined set of columns that the LLM can reason about without ambiguity. The prompt template spells out the join pattern, encoding domain knowledge about how these tables relate and what questions are meaningful. This reduces the search space for query construction while keeping each table focused and easy to inspect.

Precomputed Geographic Mapping. An important design choice is the county_nerc_mapping table, a static lookup that maps county FIPS codes to NERC subregion IDs. This mapping bridges the two spatial scales of our source data: the NERC-region-level event library (Wan, Burleyson, and Voisin 2025) and the county-level meteorology (Burleyson, Thurber, and Vernon 2023), using the 16 subregion definitions from Wan et al. (Wan et al. 2025). The LLM performs a standard relational SQL JOIN on this table at query time, guided by the prompt instruction to "join with county_nerc_mapping to ensure that only FIPS codes belonging to the same NERC_ID as the event are included." This is a relational join on a precomputed lookup, not a runtime spatial operation. This has three key benefits: (1) it ensures perfect reproducibility, as the county-to-region mapping is fixed and auditable; (2) it improves query reliability, as the join pattern is spelled out in the prompt; (3) it enhances auditability, as analysts can inspect and validate the mapping table independently of the LLM system.

Design Philosophy: Structuring for Reliability. These architectural choices reflect a broader design philosophy: rather than expecting the LLM to perform complex reasoning flawlessly, we structure the problem by encoding domain knowledge into both the data architecture and the prompt template. This approach trades flexibility for reliability: the system is highly effective for the thermal events domain but would require analogous re-structuring for other hazard types. This trade-off is acceptable in safety-critical domains where reliability matters most. Structural framing is not a limitation but an engineering choice that makes LLM-based systems viable for operational deployment. Constraining the problem lets the model focus on what it does well (natural language understanding and SQL syntax generation) while offloading tasks that proved unreliable in the GPT-4.1 implementation tested here, such as inferring multi-table join logic without explicit guidance, to deterministic preprocessing and the prompt template.

3.0 Evaluation

To validate the effectiveness and robustness of GridCoPilot, we conducted a systematic evaluation focused on two primary dimensions: (1) the correctness of the system's analytical output and (2) the reliability of the automated visualization pipeline. Our evaluation involved executing a curated set of 64 canonical queries against the Thermal Events Libraries v0 dataset.

3.1 Correctness and Quantitative Performance

The primary goal of our evaluation was to assess the accuracy of the system's end-to-end analytical response, from natural language query to SQL execution to visualization rendering. The evaluation focused on granular-level questions, which represent the most challenging use case due to their requirement for precise event identification and high-resolution spatial data retrieval. Our test suite consisted of 64 canonical benchmark queries designed to systematically probe the system's core analytical capabilities across two query patterns and two event types. Specifically, we tested “worst event” queries (e.g., “What is the worst heatwave in [NERC subregion]?”) and “worst event in a specific year” queries (e.g., “What's the worst heatwave in [NERC subregion] in year XXXX?”), each evaluated across all 16 NERC subregions for both heatwaves and coldwaves (16 subregions × 2 query types × 2 event types = 64 queries). For this evaluation, “worst event” is operationally defined as the event with the highest peak temperature within the specified NERC subregion and time period. This definition is encoded in the few-shot examples provided to the SQL Agent.

Scope of Evaluation. These 64 queries represent two well-defined query templates rather than diverse natural language formulations. The queries follow predictable patterns that align closely with the few-shot examples in our prompt, representing the core use cases for the target user workflow. Performance on more complex, ambiguous, or comparative queries (e.g., “How do 2011 and 2012 compare?” or “What defines a severe heatwave?”) was not systematically evaluated, and ad-hoc testing of such queries showed varying results.

As summarized in Table 1, GridCoPilot achieved high accuracy in the critical data retrieval phase. Across all 64 test queries following the two canonical patterns, the SQL generation achieved 100% accuracy, meaning that every generated SQL query correctly identified the target event and retrieved the appropriate data. The combination of structural framing and aligned few-shot examples produces reliable SQL generation for these core use cases. The decomposed schema, prompt-guided join logic, and explicit output format instructions appear to have eliminated the most common failure mode in NL2SQL systems for these canonical query patterns: incorrect or hallucinated queries.

However, the end-to-end accuracy (including visualization rendering) was 63/64 (98.4%). The single observed failure was due to JSON output truncation, where one coldwave “worst event” query resulted in an incomplete visualization in which the map rendered with only a subset of the affected counties because the LLM truncated the JSON response before all FIPS records were included. This failure highlights the challenge of maintaining complete structured output when queries return large result sets.

The evaluation also revealed limitations in the system's knowledge scope. A definitional query (“How do you define a heatwave?”) could not be answered meaningfully, as the system's context is strictly limited to the structured data within the database and does not include

methodological documentation or domain definitions. The system correctly responded that it could only report what data exists in the database but cannot explain how events were identified or classified. Incorporating descriptive metadata and methodology documentation into the system's retrievable context is a clear next step, one that would let the system answer not just "what" questions but also "how" and "why" questions about the underlying data.

Table 1. Evaluation results for canonical benchmark queries, showing high accuracy in event identification. The single failure was attributed to JSON output truncation.

Question Type	Event Type	Accuracy	Observed Issues
Worst event	Heatwave	16/16 (100%)	–
	Coldwave	15/16 (94%)	Incomplete visualization
Worst event in year	Heatwave	16/16 (100%)	–
	Coldwave	16/16 (100%)	–

3.2 Analysis of Observed Failure Modes

Beyond quantitative accuracy, we conducted a qualitative analysis of failure modes to understand their root causes and implications for system design. Two distinct classes of failures emerged during evaluation. Understanding these failure modes is critical for both improving the current system and informing the design of future LLM-based analytical tools.

Failure Mode 1: JSON Output Truncation. The first class, termed *JSON output truncation*, occurs when the LLM truncates the structured JSON response before all records are included. When queries return thousands of county-level records, the LLM may reach context or output token limits and truncate the JSON payload, resulting in incomplete data being passed to the visualization layer. This manifests as maps with missing counties, as shown in Figure 7, despite the SQL query itself being correct.

This failure mode motivated our empirical selection of GPT-4.1, which offers a larger context window compared to earlier models. Additionally, we implemented explicit prompt instructions ("never truncate FIPS records") to guide the model toward complete output, and a multi-strategy JSON repair pipeline that attempts to salvage partial responses through balanced-brace truncation and individual-object extraction from malformed arrays. However, for queries returning very large result sets (e.g., a 14-day heatwave affecting 500+ counties produces 7,000+ rows), truncation remains a risk. Even when the SQL is correct, output token limits can silently truncate results, a problem not unique to our system but inherent in using LLMs for structured data retrieval. Promising future mitigations include an agentic audit approach that validates response completeness, result set pagination for very large responses, or an approach that returns smaller query responses and stitches individual responses together.

Failure Mode 2: Geometry and Payload Complexity. The second failure class was related to *browser-side rendering constraints*. Relatively large visualization requests, such as those involving many counties over many days, could generate JSON payloads of 5–15 megabytes that occasionally exceeded the browser renderer's memory capacity or caused rendering timeouts. This resulted in partial animations (only some frames render), slow/frozen interfaces, or complete rendering failures. This issue was most frequent with long-duration, spatially extensive events (e.g., a 30-day coldwave affecting the entire eastern United States) and persisted even with our geometry simplification strategy in place.

This reveals a fundamental architectural tension: high-fidelity, high-resolution visualizations (which domain experts value for detailed forensic analysis) conflict with browser performance constraints. Our current mitigation (geometry simplification via Douglas-Peucker and Plotly animation frames) represents a pragmatic compromise but does not fully solve the problem for extreme cases. Alternative approaches we considered include server-side rendering (trading client performance for server load), progressive/lazy loading with viewport-based culling (only render visible map areas), or a hybrid strategy where users can request “low-resolution preview” versus “high-resolution export.” In geospatial LLM systems, payload management must be a first-class design concern, not an afterthought.

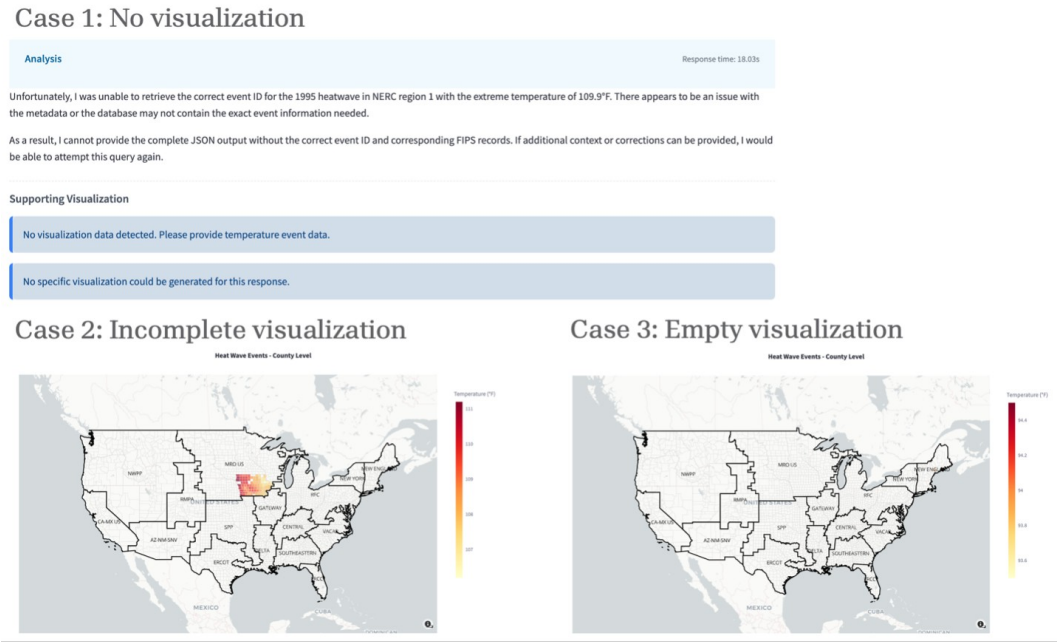

Figure 7. Examples of visualization failures observed during evaluation, including incomplete maps with missing counties, caused by JSON output truncation when result sets exceed model output limits.

3.3 Verification of Foundational Data Retrieval

While JSON truncation occasionally affected visualization, the SQL generation stage itself was fully reliable. To verify this, each query generated by the LangChain SQL Agent was manually inspected and validated against a ground-truth dataset. As illustrated in Figure 8, the system correctly executed the query process: identifying the target event from the Event_Metadata and retrieving the corresponding county-level details.

For instance, when asked for the “worst heatwave in NERC subregion 1 in 1985,” the system correctly generated a SQL query to identify Event ID H91 and its associated metadata. The retrieved data was then cross-validated and found to be identical to the ground-truth data. This 100% accuracy at the SQL generation stage confirms the effectiveness of our data transformation strategy and prompt engineering. It proves that the observed failures originate in JSON output truncation, not the core SQL generation itself.

Figure 8. Verification process for the SQL generation stage. The data retrieved by the LLM-generated SQL query (top) was cross-validated against a ground-truth dataset (bottom), confirming 100% accuracy in data retrieval.

4.0 Discussion

Our evaluation results demonstrate that careful data engineering and prompt design can achieve high reliability in the SQL generation layer of LLM-based analytical systems (100% SQL accuracy across 64 canonical queries), while also revealing that JSON output truncation remains a source of occasional failures (98.4% end-to-end accuracy). We discuss the implications for domain-specific LLM system design, the trade-offs we encountered, and pathways for generalizing this approach to other hazard domains.

4.1 Design Trade-offs and Lessons Learned

Reliability vs. Generality. Our architecture achieves high reliability by sacrificing generality. We constrain the problem space through structural framing (decomposed domain-specific schema, prompt-guided joins) rather than building a general-purpose NL2SQL system. This trade-off is evident in the contrast with prior work: general-purpose systems like NL4DV and Lida aim to handle arbitrary databases and visualization types but struggle with reliability in domain-specific contexts. By narrowing our scope to thermal event analysis with a carefully engineered schema, we achieve 100% SQL accuracy. In short, *domain-specific reliability requires domain-specific design*; there is no “one size fits all” for safety-critical applications.

Determinism vs. Flexibility. Our architecture prioritizes determinism and auditability by using deterministic visualization routines rather than LLM-generated visualization code. This design means that visualization rendering is completely reproducible: the same JSON input always produces the same map. The trade-off is reduced flexibility: adding new visualization types requires code changes rather than prompt modifications. We consider this trade-off worthwhile: in operational contexts, reproducible visualizations that occasionally fail due to JSON truncation are far less dangerous than probabilistic LLM-generated visualizations that might silently produce incorrect output.

Schema Decomposition vs. Flat Tables. A common NL2SQL strategy is to denormalize data into a single flat table to eliminate joins entirely. We took the opposite approach: separating the source long-format data into focused relational tables and guiding the LLM with prompt-based join instructions. Each table stays small and interpretable, but the LLM must execute multi-table joins correctly. The prompt guidance proved sufficient for the GPT-4.1 implementation tested here, achieving 100% SQL accuracy on canonical queries. *Prompt-guided joins over decomposed tables can be as reliable as flat-table queries* when the join pattern is well-defined and explicitly stated in the prompt. The decomposed design also preserves the ability to independently inspect and validate each table.

Visualization Fidelity vs. System Performance. The most persistent challenge is balancing visualization quality with browser rendering constraints. Domain experts expect high-resolution county-level maps for forensic analysis, but rendering 3,000+ polygons across 30+ animation frames can generate 10+ MB payloads that cause browser failures. Our geometry simplification strategy is a compromise that reduces visual precision to maintain system responsiveness. An alternative would be to maintain full-resolution geometries and accept slower render times or server-side rendering, but this would degrade the interactive experience that users value. This tension is inherent to geospatial analytics and cannot be fully eliminated; it must be actively managed through user education, progressive rendering, or tiered quality options.

4.2 Pathways for Extension: Hydropower Drought as Case Study

While our current implementation focuses on thermal events, the architectural principles we have developed are transferable to other climate hazards. However, generalization requires deliberate adaptation guided by domain experts. This subsection illustrates the extension process using hydropower drought analysis as a concrete example.

Transferable Architectural Components. Several core elements of our architecture are domain-agnostic: (1) the *LangChain SQL Agent pattern* with structured prompt engineering for verifiable SQL generation; (2) the *structural framing philosophy* of designing data schemas to reduce LLM reasoning complexity; (3) the *dual-strategy query approach* supporting both detailed forensic analysis and regional situational awareness; and (4) the *deterministic visualization pipeline* with payload management techniques. These components constitute an architectural template for domain-specific LLM systems.

Extension Roadmap: Hydropower Drought Analysis. Extending this system to hydropower drought analysis for grid infrastructure planning would require adaptation in three key areas. First, *event definitions and schema design*: hydropower droughts differ fundamentally from thermal events. They span days to seasons and sometimes to multiple years, are evaluated and managed at hydrologic region scales rather than evaluated at county levels (population impact), and require severity metrics tied to hydropower generation capacity rather than temperature thresholds. The decomposed schema would need to capture event metadata (duration, intensity, spatial extent, severity levels for generation and operational capacity) and link storage/run-of-river plants to watershed boundaries and NERC subregions. Second, *spatial pre-processing*: unlike county-to-NERC mappings used for thermal events, hydropower drought analysis requires dual mappings from hydropower plants to watershed management jurisdictions and resource adequacy regions, introducing additional spatial complexity. Third, *stakeholder engagement*: partnering with domain experts to develop meaningful query examples (e.g., “Which hydropower drought lasted the longest in the Desert Southwest?”) and decision-support visualizations.

Subject Matter Expert Engagement. Successful extension requires close collaboration with domain experts throughout development, not just as a validation phase at the end. SMEs play three essential roles: (1) *defining meaningful queries* by determining what questions domain users actually need to answer; (2) *designing event definitions and datasets* ensuring that detection and severity metrics align with domain standards; and (3) *identifying failure modes* by testing edge cases and adversarial queries. Future work should formalize this collaboration through co-design workshops and iterative evaluation cycles.

5.0 Limitations

While our system demonstrates strong performance for thermal event analysis, several limitations constrain its current applicability and should be considered when interpreting our results or planning future deployments.

JSON Output Completeness. The most significant limitation is the residual 1.6% failure rate stemming from JSON output truncation when the LLM handles large result sets. While 98.4% end-to-end accuracy represents substantial progress, it falls short of the near-perfect reliability expected in operational safety-critical systems. Silent failures, where visualizations appear correct but omit data due to truncated JSON, are particularly concerning. Current mitigation strategies (explicit “never truncate” prompt instructions, GPT-4.1's larger context window, a multi-strategy JSON repair pipeline) reduce but do not eliminate these failures for very large result sets. More robust approaches might require server-side result pagination or streaming responses.

Query Diversity and Evaluation Scope. Our evaluation uses well-formed canonical queries following two specific templates (“worst event in region” and “worst event in region in year”); performance on ambiguous, comparative, or definitional queries requires further investigation. The 100% SQL accuracy reflects these structured query patterns rather than diverse natural language formulations. Ad-hoc testing of more complex queries, such as aggregation queries (“How many heatwaves occurred in 2011?”), duration filters (“Show events lasting more than 10 days”), and definitional queries (“What defines a severe event?”), showed varying results not captured in our systematic evaluation.

Domain and Data Scope. Our evaluation is limited to a single hazard type (thermal events) using a specific curated dataset (Wan et al. 2025). The system has not been tested on other hazard types, geographic regions, or independently sourced datasets. The SQL accuracy we observe may not generalize to datasets with different schema complexity, data quality issues, or semantic ambiguities. The system's ability to handle edge cases (e.g., events spanning NERC subregion boundaries, concurrent overlapping events, data gaps) requires further investigation.

Scalability and Performance Boundaries. While our system performs well on the current dataset (tens of thousands of events, millions of observations), scalability to significantly larger datasets or higher query concurrency is unproven. The county-level detail tables, which contain granular temperature observations for each event, introduce storage overhead that may grow with additional hazard types or finer temporal resolution. Browser-side rendering constraints impose hard limits on visualization complexity; queries returning more than approximately 50,000 county-day observations (e.g., a month-long heatwave affecting the entire continental U.S.) may exceed rendering capacity even with geometry simplification. Server-side rendering or progressive loading could address this, but would require architectural changes.

LLM Model Dependency. Our system's performance is tied to GPT-4.1 (deployed via Azure OpenAI), selected specifically for its larger context window to reduce JSON truncation. As LLMs evolve, both capabilities and failure modes may shift. Improvements in structured output generation could further reduce truncation failures; conversely, changes in model behavior could introduce new failure modes. The system lacks robust version control and regression testing across LLM versions. Additionally, we have not systematically evaluated performance across different LLM providers (OpenAI, Anthropic, open-source models), which may exhibit different reliability profiles for SQL generation and structured output completeness.

User Evaluation Limitations. Our evaluation focuses on system correctness (query accuracy, data retrieval) but does not include formal user studies with domain experts. We have not measured user trust, task completion time, cognitive load, or decision quality when using the system compared to traditional workflows. User acceptance and operational effectiveness remain open questions that require ethnographic studies and controlled user experiments.

6.0 Code and Data Availability

The source code for GridCoPilot is available by request. The thermal events library used in this work is available from Wan et al. (Wan, Burleyson, and Voisin 2025) at https://doi.org/10.5281/zenodo.15306963. The county-level meteorology dataset is available from Burleyson et al. (Burleyson, Thurber, and Vernon 2023) at https://doi.org/10.57931/1960548.

7.0 Conclusion

LLM-based analytical systems can achieve high reliability in domain-specific contexts through careful data engineering and deterministic visualization pipelines. Our evaluation shows 100% SQL accuracy across 64 canonical thermal event queries, confirming that decomposed schemas with prompt-guided joins allow the model to reliably generate correct queries while avoiding common pitfalls like hallucinated joins. Beyond these technical contributions, we provide a rigorous evaluation framework that isolates failure modes and quantifies the trade-offs between generality and reliability, which is particularly relevant as organizations seek to deploy LLM-based tools in high-stakes domains where “good enough” is insufficient.

However, our results also reveal that achieving end-to-end reliability requires addressing challenges beyond query generation. JSON output truncation remains a source of occasional failures when queries return large result sets, highlighting the importance of model selection (GPT-4.1's larger context window) and explicit prompt instructions. At least for the GPT-4.1 implementation tested here, the pattern is clear: the model handles semantic understanding and translation (natural language to SQL) well, but may truncate structured output under token pressure. Using deterministic visualization routines rather than LLM-generated code eliminates an entire class of failure modes, allowing reliability efforts to focus on the SQL generation and structured output completeness phases.

Our work offers a template for domain-specific LLM deployment: prioritize structural framing over generality, use deterministic pipelines where reproducibility matters, and treat failure modes as first-class design concerns. The architectural patterns we demonstrate (LangChain SQL Agents with engineered prompts, LLM-accessible schemas, deterministic Plotly visualization) are transferable to other hazard domains and analytical contexts. With deliberate adaptation and SME collaboration, these patterns can accelerate the development of reliable LLM-based tools for safety-critical operational domains.

Key directions for future work include: (1) eliminating JSON truncation failures through result pagination or streaming responses; (2) extending to additional hazard types (wildfire, drought, flooding) to validate architectural generalizability; (3) conducting formal user studies to measure operational effectiveness; (4) incorporating uncertainty quantification and metadata retrieval for richer analytical capabilities. As LLM capabilities continue to advance, the challenge will be not what these models can do, but how to build systems that use their strengths while working around their weaknesses. This work is a step toward that goal.

8.0 References

Burleyson, C., Thurber, T., & Vernon, C. (2023). *Projections of hourly meteorology by county based on the IM3/HyperFACETS Thermodynamic Global Warming (TGW) simulations* (v1.0.0) [Data set]. MSD-LIVE Data Repository. https://doi.org/10.57931/1960548

Chaturvedi, S., Jin, S., Abhyankar, S., Thurber, T., Oikonomou, K., & Voisin, N. (2025). Grid CoPilot: A large language model (LLM) based framework for transforming long-term planning analyses. In *2025 IEEE Power & Energy Society General Meeting (PESGM)* (pp. 1–5). IEEE. https://doi.org/10.1109/PESGM52009.2025.11225574

Dibia, V. (2023). LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 113–126). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-demo.11

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *10*(2), 112–122. https://doi.org/10.3138/FM57-6770-U75U-7727

Guddanti, B., et al. (2025). Distribution system failures during extreme temperature events. *IEEE Transactions on Power Systems*. Advance online publication (In press).

Ke, X., Wu, D., Rice, J., Kintner-Meyer, M., & Lu, N. (2016). Quantifying impacts of heat waves on power grid operation. *Applied Energy*, *183*, 504–512. https://doi.org/10.1016/j.apenergy.2016.08.188

Kizilkaya, D., Sermet, Y., & Demir, I. (2025). Towards HydroLLM: Approaches for building a domain-specific language model for hydrology. *Journal of Hydroinformatics*, *27*(10), 1652–1666. https://doi.org/10.2166/hydro.2025.100

Liu, A., Hu, X., Wen, L., & Yu, P. S. (2023). A comprehensive evaluation of ChatGPT's zero-shot text-to-SQL capability. *arXiv*. https://doi.org/10.48550/arXiv.2303.13547

Luo, Y., Tang, N., Li, G., Tang, J., Chai, C., & Qin, X. (2022). Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, *28*(1), 217–226. https://doi.org/10.1109/TVCG.2021.3114848

Narechania, A., Srinivasan, A., & Stasko, J. (2021). NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), 369–379. https://doi.org/10.1109/TVCG.2020.3030378

Panteli, M., & Mancarella, P. (2015). Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies. *Electric Power Systems Research*, *127*, 259–270. https://doi.org/10.1016/j.epsr.2015.06.012

Pourreza, M., & Rafiei, D. (2023). DIN-SQL: Decomposed in-context learning of text-to-SQL with self-correction. In *Advances in Neural Information Processing Systems 36* (pp. 36339–36348). https://doi.org/10.52202/075280-1577

Rajkumar, N., Li, R., & Bahdanau, D. (2022). Evaluating the text-to-SQL capabilities of large language models. *arXiv*. https://doi.org/10.48550/arXiv.2204.00498

Scholak, T., Schucher, N., & Bahdanau, D. (2021). PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 9895–9901). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.779

Wan, H., Burleyson, C., & Voisin, N. (2025). *Heat wave and cold snap event library under various technical choices for NERC subregions in the conterminous U.S. (1980–2024)* [Data set]. Zenodo. https://doi.org/10.5281/zenodo.1530696

PNNL-39202

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov