ChatGrid: Power Grid Visualization Empowered by a Large Language Model

Sichen Jin*
School of Interactive Computing
Georgia Institute of Technology

Shrirang Abhyankar†
Electricity Infrastructure and Buildings Division
Pacific Northwest National Laboratory

2024 IEEE Workshop on Energy Data Visualization (EnergyVis) | 979-8-3503-7923-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/EnergyVis63885.2024.00007

Abstract

This paper presents a novel open system, *ChatGrid*, for easy, intuitive, and interactive geospatial visualization of large-scale transmission networks. ChatGrid uses state-of-the-art techniques for geospatial visualization of large networks, including 2.5D map views, animated flows, hierarchical and level-based filtering and aggregation to provide visual information in an easy, cognitive manner. The highlight of ChatGrid is a natural language query based interface powered by a large language model (ChatGPT) that offers a natural and flexible interactive experience whereby users can ask questions and ChatGrid provides responses both in text and visually. This paper discusses the architecture, implementation, design decisions, and usage of large language models for ChatGrid.

Index Terms: Power Grid, Visualization, LLM, NL2VIS

1 Introduction

With the ongoing modernization efforts to enhance reliability, resiliency, flexibility, affordability, and equity, the grid is undergoing a transformation with new technologies, processes, and thereby also increasing the complexity of planning and operation. In this context, the operators, planners, and researchers heavily rely on advanced analytic techniques to understand and predict how the grid operates. Drawing insights and inferences from these analytics through visualization is essential for quickly identifying patterns, such as power spatial distributions. As such, power grid visualization is extremely valuable for power utilities, researchers, and policymakers, enabling them to extract patterns, monitor and optimize the performance of modern power grids in an increasingly dynamic and efficient manner. However, visualization of large-scale geospatial networks, such as the power grid, is challenging due to several aspects from large-scale geography, visual clutter, cognitive overload, and even the learning curve of the visualization tool.

Over the years, researchers have developed different visualization techniques starting with the seminal work of Thomas Overbye [29, 25, 26, 28] on geospatial visualization of transmission networks, dynamic line flows, and contouring. [1] and [2] presented displaying the 2D grid information (such as power flows and voltages) through a third dimension. For comprehensive surveys of different visualization techniques used, we refer the reader to [31, 13].

In this paper, we present a state-of-the-art novel visualization tool, named *ChatGrid*, for visualization of transmission networks. Our emphasis and motivation in developing ChatGrid are (a) developing visualization for large-scale transmission networks minimizing the "visual clutter", (b) exploring the usage of large language models, such as ChatGPT, to assist in answering analytic queries, and (c) visually displaying the answer from the large language model. Figure 1 shows a sample of the ChatGrid interface with its map-based visualization and the natural language interface for data queries. In this sample, the user asks ChatGrid for information on generatoring stations with remaining capacity (headroom) greater than 100 MW and ChatGrid returns this information in text and displays the locations of those generation sources on the map.

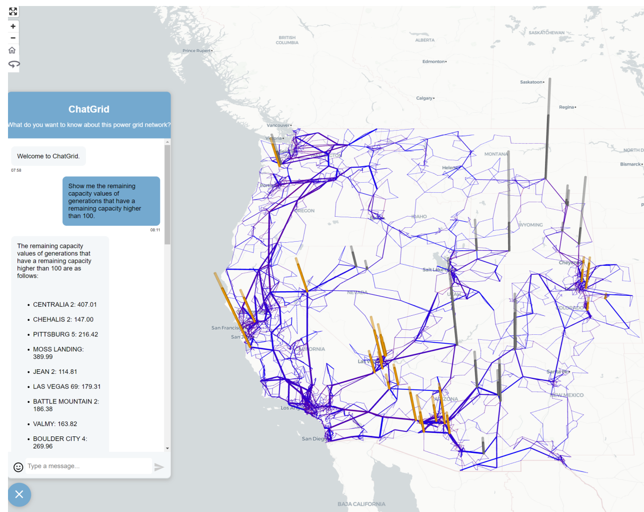

Figure 1: ChatGrid interface displaying the visualization and query interface. Queries asked by users are responded through both text and visualization. The vertical bars represent the generation sources that have a remaining capacity greater than 100 MW.

The developed ChatGrid framework is a part of the Exascale Grid Optimization (ExaGO) [4, 3], and is currently used for visualizing the AC optimal power flow (ACOPF) results from ExaGO. We have tested ChatGrid to visualize ACOPF results from ExaGO on transmission grids as large as the combined synthetic U.S. grid cases [11]. We note here that though ChatGrid is part of ExaGO, it can be used for visualizing power flow or AC optimal power flow results from other analysis tools through creating appropriate input files. In this paper, we first describe the visualization framework of ChatGrid including its different features. Next, we detail its ChatGPT-based natural language interface and discuss the design choices we made. Lastly, we summarize the lessons learned and the next steps.

2 Related Work

2.1 Power Grid Visualization

Power grid networks are complex data structures that require multifaceted analysis, including multivariate analysis, geospatial analysis, network analysis, and outlier detection. Previous work has examined various visualization techniques for power grid networks [24, 13, 31] and there are commercial products like Powerworld Simulator [29] and RTDMS [9] designed for power grid visual analytics. Among typical power grid visualizations, high-dimension multivariate visualization has been widely adopted in power grid

*sichenjin@gatech.edu
†shri@pnnl.gov

979-8-3503-7923-5/24/$31.00 ©2024 IEEE
DOI 10.1109/EnergyVis63885.2024.00007

analysis tasks, such as using scatterplot matrix [5] for consumer demand estimation, parallel coordinates for evaluating factors affecting heating demand [5], and dendrogram to show epicentric event propagation[7].

Glyphs are another common way of encoding multivariate data in the power systems, representing information such as voltage limits [28], transmission load percentage [28], bus voltage stability [45], faults detection [43]. However, as the scale and network size of the visualization increase, visual clutter can become problematic, making glyphs less distinguishable. To address this, some work extends 2D visualization into a 3D space [25] to broaden the visual encoding bandwidth, which has been shown to improve solution time and reduce errors in power grid monitoring tasks due to more salient visual encoding and less visual clutter. In addition to multivariate visualization, geographical maps with overlaid topology are often the primary presentation of location and topology information. Some approaches include additional geographic layers to display temperature [27] and regional voltage profiles [16, 33] using contour maps. There are also works [41, 40, 18] that propose layout algorithms to preserve both geographic layout and topological relationships for more efficient grid cluster identification (e.g., synchronous nodes).

2.2 NL2VIS

In the rapidly advancing field of Natural Language Processing (NLP), the utilization of natural language as an intermediary interaction interface for visualization provides opportunities to reduce the learning curve of operating dataflow systems [34]. Works in this area include automatic visualization creation [39, 10, 30], conversational interfaces for visualization [8, 12] that support data queries such as retrieving values, deriving values, determining range, filtering, etc[14, 36, 38]. However, most previous works are built upon design rules formulated according to data types and utilize grammar and lexical parsing techniques [42, 15, 20, 44, 23]. The predefined template mapping for attribute and task inference can limit the tool's generality, particularly when dealing with underspecified queries that may include synonyms and similar challenges. Recent work attempts to overcome these limitations by integrating deep learning models, as exemplified by projects such as ADVISor [19], ncNet [21], RGvisNet [35], Chat2VIS[22]. These deep learning-based approaches leverage visualization grammars or programming codes, such as Vega-lite [32], Vega-zero [21], Python scripts [22] to reframe the problem of translating natural language to visualization as one of generating visualization grammar code or expressions. We build on previous work in NL2VIS by extending its application to the context of power grid management and making massive and complex power grid datasets more accessible and actionable for power grid operators.

3 ChatGrid - Visualization

The input of ChatGrid will be a GeoJSON file with the substations/buses as "points" and branches as "links". The substation/buses include generation and load information, and the branches include information on the flows, capacities, etc. This input GeoJSON file can be created by ExaGO [4, 3] or externally.

The visualization is implemented using open-source libraries, including Deck.gl and Chart.js. ChatGrid's visualization relies heavily on the Deck.gl library [37], a WebGL-powered framework that provides fast rendering capabilities for handling large power grid networks. The ChatGrid visualization is detailed through the following views in the next subsections:

3.1 Generation View Layer

The generation view layer (Figure 2) chooses a 2.5D projection visualization providing an additional visual dimension beyond color, size, or shape on a 2D map to reduce visual clutter. Tilting the map with a small angle allows users to effectively discern variations in bar height and make assessments of its value. The color of the bar indicates the generation type (e.g., wind, solar, coal, etc.), and the height displays the power generated by it. Users can toggle on a capacity layer to visualize the generation capacity, displayed as a semi-transparent bar overlaying the power generation bar. The exceeded semi-transparent bar indicates the remaining capacity. We chose a transparency level that maintains clarity while allowing the underlying bar colors to remain distinguishable. Additionally, the color scheme is selected to ensure sufficient contrast between the different layers to reduce color interference. When the generation layer is activated, a doughnut chart displays the proportion of power generated by each generation type. Users can select specific types of generators by clicking on corresponding sectors of the doughnut chart. For instance, by hiding other types of generations, it becomes apparent that most hydroelectric generators are located near the coast or close to water resources.

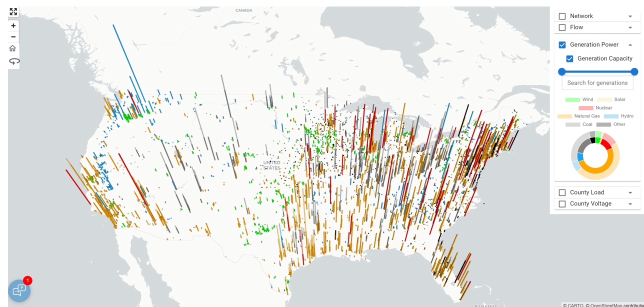

Figure 2: The generation view layer represents the capacity and power generated of each generation using the height of bars within a 2.5D projection.

3.2 Network View Layer

The network layer (Figure 3) visualizes buses and transmission lines on a base map. Each point represents a substation, which contains several buses and generations. A transmission line connects two substations, with color encoding the load ratio passing through it (i.e., the amount of actual flow/ line capacity), ranging from 0 to 100% and line width encoding the kilovolt (KV) level of the transmission line. The network layer supports searching by substations and/or transmission line names. In addition, users can apply filters to visualize the network at a specific KV level or KV level range.

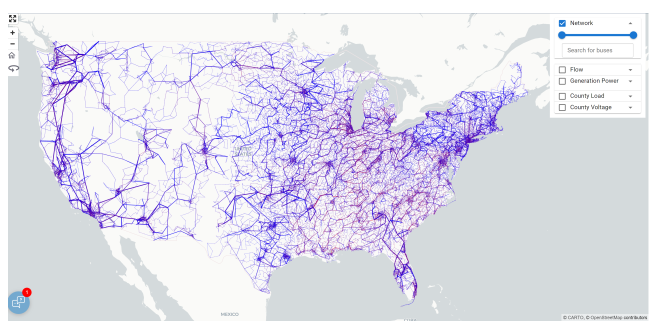

Figure 3: The network layer visualizes buses and transmission lines on a map with line color representing load ratio and line width representing voltage level.

The network view includes a flow layer (Figure 4) to display line flows as animated dashed lines. In the flow layer, color represents the flow ratio, which is the ratio of actual flow to line capacity, and the line width shows the amount of flow. The direction of power transmission is visually conveyed through the movement of the dashed lines, with the dashes moving along the line from the power source to the destination. The flow layer supports aggregation so as the zoom level changes, the granularity of flow visualization adjusts accordingly. Additionally, users can filter transmission lines by loaded percentage with the flow ratio slider.

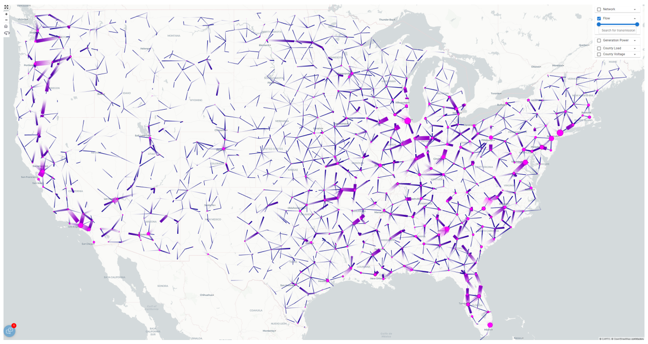

Figure 4: The flow layer shows the power transmission direction with animated dash lines. Substations nearby are aggregated to reduce visual clutter.

3.3 Load View Layer

The county load layer shows the total load of buses within each county with a continuous gray to red color scale to indicate low-load counties to high-load counties, respectively. As seen in figure 5, the load hot spots are usually big city urban areas such as Los Angeles, Chicago, New York, Houston, etc. Clicking on a county of interest like Los Angeles County, the map will zoom into it and provide detailed information through a tooltip. Utilizing the load slider filter, one can filter the visualization to look for counties that have a specific range of electricity consumption.

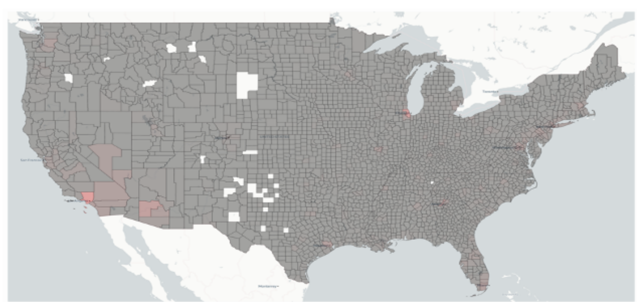

Figure 5: The county load layer shows the total load of buses within each county with a continuous color scale that transitions from gray to red to indicate low-load counties to high-load counties, respectively.

4 ChatGrid - a Large Language Model Interface

As discussed in the previous section, the visualization interface allows users to interact with the data through some predefined filter widgets such as sliders. However, using such widgets has several drawbacks - (a) displaying a large number of attributes increases the number of widgets or increases the complexity of the visualization, (b) data interactivity and exploration are limited to the functionality of these widgets, and, lastly, (c) there is a learning curve to learn how to operate these widgets (e.g. understanding when to apply union versus intersection).

To tackle the above problems of widget interactions, ChatGrid incorporates a natural language interface between user queries and visualization outputs to support a more flexible and natural interaction experience. With this approach, users can pose queries using natural language and receive updated visualizations and text summaries as responses.

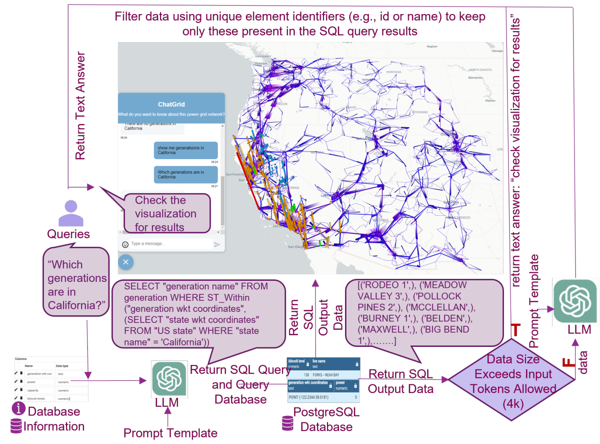

Figure 6: The system architecture of ChatGrid with a spatial query as the use case.

There were two considerations/challenges in designing this interface. First, LLMs on their own, without the aid of augmentation tools like search engines, do not have access to knowledge beyond their training corpus. So, they do not have information on our local dataset used for visualization. Second, the response from ChatGPT needs to be also converted to a visual display. The architectural design and the flow of information from the user query to ChatGrid to the LLM and back to the user is shown in Figure 6. We break down this workflow and discuss key components and design considerations in the next subsections.

4.1 Connecting LLMs to Local Datasets

There are two common ways to adapt LLMs to understand personal datasets. The first is to fine-tune the model's parameters by training it on specific datasets. The training or fine-tuning process involves providing LLMs with examples from the local datasets, such as example question and answer pairs. The model will then learn from this data and adapt its language understanding and generation capabilities to be contextually relevant to the local datasets. The second option involves using a combination of external tools and LLMs to generate answers to queries. This is usually achieved by leveraging prompt strategies and API to external tools. For instance, to compose a prompt instructing LLMs to create executable API commands, execute these commands using external tools, subsequently process results, and return the outputs.

Comparing the two approaches, retraining the model enhances the LLMs' knowledge of specific topics, particularly when textual inputs and outputs are available. However, power grid data is typically stored as structured network datasets. For our data query task, a combination of reasoning (e.g., calculation) and actions (e.g., searching, visualizing) is required beyond mere conceptual understanding. Additionally, power grid data can include sensitive information that LLMs cannot access. Therefore, we chose to integrate LLMs with API-enabled external tools, such as LangChain and SQL databases.

Specifically, we used LangChain, an auxiliary framework to set up steps and actions to interact with ChatGPT, for prompt template configuration and output parsing. In addition, all the local data is stored in an SQL database. We chose PostgreSQL because its spatial extension, PostGIS, provides a comprehensive set of geographic operations for spatial data queries.

Table 1: Performance Evaluation across Query Types. A ✔ sign implies the returned results are partially correct (i.e., a subset of the correct answer is returned). A ✓ sign implies the returned results are not always correct. A ✅ sign implies the returned answers are always correct.

Query Type		Example Input Query	SQL
Data Lookup	name constraints	How much power is generated by the generation ***"RANDLE"*** ?	✅
	numeric constraints	List generations with capacity ***higher than 600***.	✅
	categorical constraints	List all the ***wind*** generations	✅
Value Derivation	aggregates	How many wind generations are there ***in total***?	✅
	derive new attributes and values from existing ones	List transmission lines that are ***loaded more than 50%***. (given the actual flow and capacity of each line).	✅
	spatial relationship	Which generations are ***in California***?	✓
Logical Inference	or, and	List all the wind generations ***and*** generations with a capacity higher than 1300.	✅
	negation	List names of all generations ***except for*** wind generations.	✅
	nested	List the top 5 coal generations ***that*** generate the most power.	✅
Semantic Inference	subjective description	Return the most ***robust*** generations.	✔
	spelling error	List all the ***caol*** generations.	✅
	synonyms	Show me all the generations with a ***storage limit*** higher than 600.	✅

The overall system architecture is shown in Figure 6. When a user inputs the query "Which generations are in California" into the popup chat window, ChatGrid uses a predefined template to construct a prompt by combining the query with database information, including table names, attribute names, attribute types, and descriptions. This assembled prompt is then forwarded to ChatGPT for processing. To fetch relevant information, ChatGPT first identifies that there is a table named "generation," but it lacks the information about "state"; instead, it contains only the geographic coordinates (latitude and longitude) of each generation. Subsequently, ChatGPT detects another table named "US state" which stores the geographical boundaries of all states. ChatGPT then generates a SQL query that calls a PostgreSQL function to determine which generations are located within the geographical boundaries of California, based on their coordinate information. ChatGrid uses this SQL query to query the database and send the output data to be visualized and summarized.

4.2 Performance Evaluation across Query Types

We conducted performance evaluations across four types of data queries: data lookup, value derivation, logical inference, and semantic inference on a synthetic US Western power grid network [11]. This synthetic network comprises 12,709 transmission lines, 10,000 buses, 4,762 substations, and 849 generating substations. We follow previous work of a text-to-SQL benchmark that builds a taxonomy that covers different classes of queries [17]. Data lookup queries are queries where the desired answer is readily present in the provided dataset, while value derivation queries require some form of calculation or additional processing to obtain the answers. Logical inference queries are queries that involve logical operations, such as "and, or, negation". Semantic inference queries are queries that include inaccuracies or ambiguities expression in the queries. Table 1 lists example queries and results. The check icon in half green (✔) implies the returned results are partially correct (i.e., a subset of the correct answer is returned). The dashed check icon (✓) means the returned results are not always correct, as variations such as rephrasing can affect the results. The square green check icon (✅) implies the returned answers are always correct regardless of rephrasing. The experimental setup includes ChatGPT-Turbo 3.5 with a temperature setting of 0, and LangChain@0.0.233.

Overall, ChatGrid exhibits high accuracy in addressing all four types of queries and successfully returns all the data points that match specific query conditions in most cases. In addition, ChatGrid's performance is not limited by the data size as it only requests SQL query returns given database schemas and does not necessarily need to feed the whole data into the LLM.

On the other hand, there are certain factors that would affect the performance of ChatGrid. First, for value derivation queries, we notice the performance also depends on the supported functionalities of the selected database, especially when complex calculation is involved. In our example, we choose PostgreSQL as our database. It has an extension called PostGIS, which offers an extensive array of functions designed for handling spatial data queries. Therefore, ChatGrid supports responding to geospatial queries that involve various spatial operations such as containment and intersection but these queries might not succeed in other databases that do not support spatial queries. However, ChatGrid can become less stable when dealing with these complex queries that involve external functions (e.g., determining geometries' relationships), where rephrasing queries could yield different outputs.

Second, ChatGrid mostly relies on table descriptions and attribute names to perform semantic inference, such as subjective descriptions, spelling errors, and synonyms that appeared in the queries. Therefore, providing detailed and necessary descriptions of the database can help it develop an understanding of the dataset and deal with synonyms and other semantic inference queries.

4.3 Visualize Output

Our next step was to make the visualization respond to the updates of the SQL agent's outputs. We formulated a set of rules that function as APIs that define the data flow and processes and visualization choices. It's important to note that these rules are highly system-specific and, therefore can constrain generability and automation of the system. Our strategy for maintaining automation integrity involves using universal identifiers, specifically element IDs, to establish a link between a data entry point and a visual element. We explicitly instruct the LLM to generate SQL queries that retrieve data IDs for each query in the prompt template.

After receiving the data entries and their IDs matching the query conditions from the PostgreSQL database, we use these IDs to filter visual elements to be shown to present the query results. In addition, the visualization will automatically activate layers containing elements in the returned outputs and adjust the map view to encompass the bounding box of the displayed elements.

At the same time, the data entries from the query results will be forwarded to the LLM, where a text summary of the observations will be requested and posted in the conversation window. It's important to note that this is the only time when we send original data to the LLM. For projects involving sensitive data, an alternative approach can be used to produce other formats of results, such as local processing to generate tabular data entries.

5 Conclusions and Next steps

In this work, we have demonstrated a novel modern power system visualization framework, ChatGrid, empowered with an LLM.

Through enhanced 2.5D projection, dynamic maps, hierarchical filtering, and county-based heat maps, ChatGrid provides functionality for quick visual analytics. Through its natural language interface, the framework pushes the visualization paradigm from “what you see is what you select” to “what you see is what you ask”, offering a more flexible approach to visual data analytics that serves as an alternative to traditional widget interactions. ChatGrid currently supports most low-level components of analytic activities in information visualization, including retrieving values, filtering, computing derived values, finding extremums, sorting, determining ranges, and characterizing distributions [6]. However, the NL2VIS component needs to be further automated to support other tasks like finding anomalies, clustering, and correlation. Our results indicate that using ChatGPT to answer queries results in satisfactory answers in most cases, but not all. This needs further exploration. As with most visualization interfaces, a human factor assessment study is a logical next step to improve ChatGrid.

References

[1] Visualizations for power system contingency analysis data. *IEEE Transactions on Power Systems*, 19(4), 2004. 1

[2] Three-dimensional visualization and animation for power systems analysis. *Electric Power Systems Research*, 79(12), 2009. 1

[3] S. Abhyankar, S. Peles, T. Becejac, J. Holzer, A. Mancinelli, and C. Rutherford. Exascale grid optimization (exago) toolkit: An open-source high-performance package for solving large-scale grid optimization problems. *arXiv preprint arXiv:2203.10587*, 2022. 1, 2

[4] S. Abhyankar, S. Peles, A. Mancinelli, C. Rutherford, B. Palmer, J. Litzinger, W. Perkins, S. A. Sakin, J. Macam, R. Danehy, and N. Koukpaizan. Exascale Grid Optimization Toolkit. https://github.com/pnnl/ExaGO, 2023. 1, 2

[5] S. Ali, C. Mehmood, B. Khan, M. Jawad, U. Farid, J. Jadoon, M. Ali, N. Tareen, S. Usman, M. Majid, et al. Stochastic and statistical analysis of utility revenues and weather data analysis for consumer demand estimation in smart grids. *Plos one*, 11(6):e0156849, 2016. 2

[6] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 111–117. IEEE, 2005. 5

[7] A. Arunkumar, A. Pinceti, L. Sankar, and C. Bryan. Pmu tracker: A visualization platform for epicentric event propagation analysis in the power grid. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1081–1090, 2022. 2

[8] S. Bieliauskas and A. Schreiber. A conversational user interface for software visualization. In *2017 ieee working conference on software visualization (vissoft)*, pp. 139–143. IEEE, 2017. 2

[9] H. Chen, J. Mo, U. Kothapa, B. Bhargava, and J. Dyer. Development of an on-line synchrophasor wide area dynamics monitoring platform. In *2014 IEEE PES T&D Conference and Exposition*, pp. 1–5. IEEE, 2014. 1

[10] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE transactions on visualization and computer graphics*, 26(1):906–916, 2019. 2

[11] A. B. et. al. Texas A&M University Electric Grid Datasets. https://electricgrids.engr.tamu.edu/, 2023. 1, 4

[12] E. Fast, B. Chen, J. Mendelsohn, J. Bassen, and M. S. Bernstein. Iris: A conversational agent for complex tasks. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pp. 1–12, 2018. 2

[13] M. T. Fischer and D. A. Keim. Towards a survey of visualization methods for power grids. *arXiv preprint arXiv:2106.04661*, 2022. 1

[14] S. Fu, K. Xiong, X. Ge, S. Tang, W. Chen, and Y. Wu. Quda: Natural language queries for visual data analytics. *arXiv preprint arXiv:2005.03257*, 2020. 2

[15] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th annual acm symposium on user interface software & technology*, pp. 489–500, 2015. 2

[16] K. M. Gegner, T. J. Overbye, K. S. Shetye, and J. D. Weber. Visualization of power system wide-area, time varying information. In *2016 IEEE Power and Energy Conference at Illinois (PECI)*, pp. 1–4. IEEE, 2016. 2

[17] O. Gkini, T. Belmpas, G. Koutrika, and Y. Ioannidis. An in-depth benchmarking of text-to-sql systems. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 632–644, 2021. 4

[18] A. Godwin. Paths through spatial networks. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pp. 31–34. IEEE, 2022. 2

[19] C. Liu, Y. Han, R. Jiang, and X. Yuan. Advisor: Automatic visualization answer for natural-language question on tabular data. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, pp. 11–20. IEEE, 2021. 2

[20] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th international conference on data engineering (ICDE)*, pp. 101–112. IEEE, 2018. 2

[21] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin. Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):217–226, 2021. 2

[22] P. Maddigan and T. Susnjak. Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models. *IEEE Access*, 2023. 2

[23] A. Narechania, A. Srinivasan, and J. Stasko. 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, 2020. 2

[24] D. V. Nga, O. H. See, C. Y. Xuen, L. L. Chee, et al. Visualization techniques in smart grid. *Smart Grid and Renewable Energy*, 3(03):175, 2012. 1

[25] T. Overbye, R. Klump, and J. Weber. Interactive 3d visualization of power system information. *Electric Power Components and Systems*, 31(12):1205–1215, 2003. 1, 2

[26] T. Overbye, A. Meliopoulos, D. Wiegmann, and G. Cokkinides. Visuaization of power system components, 2005. 1

[27] T. Overbye, F. Safdarian, W. Trinh, Z. Mao, J. Snodgrass, and J. Yeo. An approach for the direct inclusion of weather information in the power flow. 2023. 2

[28] T. J. Overbye, A. Meliopoulos, D. Wiegmann, and G. Cokkinides. Visualization of power systems and components. *PSERC Publication*, (05-65), 2005. 1, 2

[29] T. J. Overbye and J. D. Weber. New methods for the visualization of electric power system information. In *IEEE Symposium on information visualization 2000. INFOVIS 2000. Proceedings*, pp. 131–16c. IEEE, 2000. 1

[30] C. Qian, S. Sun, W. Cui, J.-G. Lou, H. Zhang, and D. Zhang. Retrieve-then-adapt: Example-based automatic generation for proportion-related infographics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):443–452, 2020. 2

[31] M.-A. Sanchez-Hidalgo and M.-D. Cano. A survey on visual data representation for smart grids control and monitoring. *Sustainable Energy, Grids and Networks*, 16:351–369, 2018. 1

[32] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics*, 23(1):341–350, 2016. 2

[33] M. Sheikholeslami. *Visualization for Power System Analysis and Control*. PhD thesis, State University of New York at Buffalo, 2017. 2

[34] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang. Towards natural language interfaces for data visualization: A survey. *IEEE transactions on visualization and computer graphics*, 2022. 2

[35] Y. Song, X. Zhao, R. C.-W. Wong, and D. Jiang. Rgvisnet: A hybrid retrieval-generation neural framework towards automatic data visualization generation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1646–1655, 2022. 2

[36] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation

and communication. *IEEE transactions on visualization and computer graphics*, 25(1):672–681, 2018. 2

[37] Uber. Deck.gl. https://deck.gl/, 2023. 2

[38] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE transactions on visualization and computer graphics*, 26(1):895–905, 2019. 2

[39] Y. Wang, H. Zhang, H. Huang, X. Chen, Q. Yin, Z. Hou, D. Zhang, Q. Luo, and H. Qu. Infonice: Easy creation of information graphics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2018. 2

[40] P. C. Wong, H. Foote, P. Mackey, G. Chin, Z. Huang, and J. Thomas. A space-filling visualization technique for multivariate small-world graphs. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):797–809, 2011. 2

[41] P. C. Wong, K. Schneider, P. Mackey, H. Foote, G. Chin Jr, R. Guttromson, and J. Thomas. A novel visualization technique for electric power grid analytics. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):410–423, 2009. 2

[42] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2015. 2

[43] R. Xie, F. Wang, W. Li, Y. Huang, W. Peng, S. Zhang, and W. Zheng. Towards progressively detecting faults in a large power system: A visual analytics approach. *IEEE Transactions on Industry Applications*, 2023. 2

[44] B. Yu and C. T. Silva. Flowsense: A natural language interface for visual data exploration within a dataflow system. *IEEE transactions on visualization and computer graphics*, 26(1):1–11, 2019. 2

[45] T. Zhang, Q. Wang, L. Lin, J. Xia, X. Xu, Y. Huang, X. Luo, W. Zheng, and W. Chen. Wavelines: towards effective visualization and analysis of stability in power grid simulation. *Frontiers of Computer Science*, 15(6):156704, 2021. 2