GridMind: An Agentic Workflow for Weather-Driven Power Grid Risk Assessment

D. Adrian Maldonado
maldonadod@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Jonghwan Kwon
kwonj@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Sunash Sharma
sunash.sharma@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Hongwei Jin
jinh@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Stefano Fenu
sfenu@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Emil M. Constantinescu
emconsta@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

David Sehloff
dsehloff@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Parfait Gasana
pgasana@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Duane Verner
dverner@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Kibaek Kim
kkim@anl.gov
Argonne National Laboratory
Lemont, Illinois, USA

Abstract

We introduce GridMind, an agentic workflow with a natural-language interface for weather-driven power-grid risk assessment. Scientific agents that call long-running simulation tools must determine not only which tool to use, but whether intermediate results from prior turns are still valid for the current request. GridMind addresses this problem while orchestrating outage generation, AC optimal power flow, contingency analysis, and cascade simulation. Each stage writes outputs under a content hash of its inputs, and the orchestrator advances by checking required artifacts rather than action history. As a result, a follow-up query can change one analysis parameter without forcing the agent either to rerun the whole workflow or to trust intermediates produced under incompatible inputs. We test the agentic workflow on a simulated hurricane risk-assessment scenario using a synthetic 2000-bus grid case set in Texas. GridMind executes the full workflow and reuses artifacts across follow-up queries that change cascade scope, solver choice, and forecast initialization.

CCS Concepts

• **Computing methodologies** → **Intelligent agents**; **Multi-agent systems**; *Neural networks*; • **Applied computing** → **Engineering**.

Keywords

LLM agents, power grid, risk assessment, workflow orchestration

1 Introduction

Extreme weather is a leading cause of large-scale power outages in the United States. For instance, Winter Storm Uri and Hurricane Beryl each left millions without power and caused billions in damage, and the frequency of such events is increasing [13]. Grid operators need to assess infrastructure vulnerability before an event arrives, which means executing risk analysis workflows fast enough to inform pre-event operational decisions.

Planning for the impact of weather events on the power grid requires coordinating analyses across distinct domains. Assessing which transmission infrastructure is exposed to an event, how that exposure translates into equipment failures, and how those failures propagate through the network involves tools from meteorology and power systems that are developed independently, maintained by different organizations, and rarely interoperable. Orchestrating this analysis manually is slow and error-prone, which limits the number of scenarios that can be evaluated and the lead time available for operational decisions.

Interactive agents can reduce this manual burden, but they introduce a state-management problem. Grid studies produce intermediate artifacts that are large, expensive to recompute, and tied to specific inputs: forecast initialization, outage schedule, network case, solver choice, time window, and query parameters. When a user changes one of these inputs in a follow-up request, the agent must determine which prior artifacts remain valid and which stages must be rerun. A record of actions previously taken is not enough, because the same action may have been executed under different inputs.

We present **GridMind**, an agentic workflow with a natural-language interface for weather-driven grid risk analysis. The workflow spans outage generation, AC optimal power flow (ACOPF), contingency analysis, and cascading outage simulation. The workflow is heterogeneous by construction: it ingests multiple multi-hour wind forecasts from AERIS [8], a weather foundation model; uses a probabilistic outage model [7] to convert weather exposure into candidate asset failures; leverages LUMINA [11], an ACOPF foundation model, to evaluate many network contingencies; and

dispatches selected scenarios to EPFast [15], a physics-based cascade simulator that resolves contingency scenarios into load loss. These stages produce large, typed data products, from per-line wind time series to contingency rankings and cascade outputs. GridMind uses natural language to specify and revise this workflow across turns; content-hashed artifacts make that interaction robust. Each stage writes its outputs under a hash of its inputs, and the orchestrator checks whether the required artifacts exist in a persistent store rather than consulting action history. This gives the agent a well-defined termination condition and makes partial reuse automatic: when only one upstream input changes, only the affected stages rerun. For example, changing the number of cascade scenarios changes only the cascade signature, whereas changing the forecast initialization changes the weather-derived line-wind artifact and invalidates all downstream stages.

Our contributions are: (1) GridMind, a natural-language workflow that connects ensemble weather forecasts to outage generation, ACOPF, contingency analysis, and cascade simulation; (2) a content-hash-based orchestration mechanism for interactive scientific agents, where artifact identity rather than action history determines which stage executes next; and (3) an evaluation on the Texas 2000-bus system, a medium-sized grid test case, using live weather, ACOPF, contingency, and cascade backends to show full-workflow execution, correct invalidation under changes to cascade scope, solver choice, and forecast initialization, and a 60% reduction in stage executions across a seven-query session.

2 Related Work

LLM-based agents are beginning to enter power-system analysis, where one challenge is translating user intent into valid domain-tool calls. Grid-Agent [19] combines LLM reasoning with validation and rollback for violation detection and remediation. X-GridAgent [18] uses a hierarchical planning, coordination, and action architecture for natural-language grid-analysis queries. PowerDAG [2] improves distribution-grid analysis through adaptive retrieval and just-in-time correction of tool-use errors. These systems show how language models can mediate between users and grid-analysis software, but they largely remain within grid-internal analyses.

A parallel line of work connects LLMs to scientific software more broadly. ChemCrow [4], MDCrow [5], and AutoSolvate-Agent [6] demonstrate that language models can plan and execute multi-step computational tasks in chemistry, molecular dynamics, and quantum chemistry. In these settings, the agent coordinates computations whose intermediate products shape later decisions. This is directly relevant to weather-driven grid-risk assessment, where forecast extraction, outage generation, contingency screening, and cascade simulation produce line-wind series, outage schedules, ranked contingencies, and load-loss outputs. The interaction is therefore both conversational and data-intensive.

Scientific workflow systems have long treated intermediate products as first-class objects. Snakemake [12] represents analyses as user-defined rules over declared inputs, outputs, parameters, and software environments, avoiding reruns when products are already up to date. AiiDA [14], signac [1], and Workflow Run RO-Crate [10] manage data spaces and record provenance over calculations, inputs, outputs, code, and execution metadata. More recently, FlowMesh [16] uses lineage and content-addressable storage to deduplicate work across composable LLM workflows. Together, these systems point to an artifact-centered view of computation: the state of a scientific analysis is not only the sequence of actions that has occurred, but the set of data products produced under particular inputs.

GridMind combines these threads for multi-turn grid-risk analysis: it exposes domain tools through natural language, coordinates heterogeneous scientific software, and treats intermediate products as persistent artifacts. Unlike rule-based systems such as Snakemake, where the user declares rules with typed inputs and outputs and reruns are triggered by file modification time, GridMind infers required stages from natural language and uses content hashes of stage inputs to determine validity, so the set of required stages may differ between consecutive queries in the same session.

3 System Design

3.1 Overview

GridMind is implemented as a LangGraph hub-and-spoke graph [9]. A central policy node, the orchestrator, uses an LLM call to classify the natural-language request into a structured set of required action types, then deterministically enforces dependency order and injects missing prerequisites. The orchestrator then executes the first unsatisfied action in dependency-safe order. For example, if a user asks to compute the risk of cascading on a given day, GridMind infers that a weather forecast and a set of outages are required first and that contingency analysis is needed before cascade simulation if no ranked scenarios exist. Five domain spokes carry out the analysis, corresponding to weather forecasts, outage generation, ACOPF, contingency screening, and cascade simulation (Fig. 1). The LLM resolves what the user wants; the artifact-signature graph resolves what needs to run. This division lets GridMind support context-dependent follow-up requests while keeping execution grounded in explicit artifact dependencies: the LLM output is validated against a fixed action vocabulary and reordered before any stage executes, so an invalid or hallucinated action cannot reach a backend.

3.2 Pipeline Stages

The *weather* spoke interfaces with AERIS [8], a weather foundation model, and obtains forecasts across multiple initial conditions and ensemble members for the desired analysis period. For hurricane scenarios, AERIS provides forecasts at multiple issue times, with an ensemble of members for each issue time. These members represent alternative forecast trajectories for the storm and its associated wind fields. The stage extracts wind speed time series at selected locations close to the network infrastructure. The *outage* spoke uses a stochastic outage generator conditioned on wind intensity to translate per-line wind speed distributions into candidate outage schedules with associated failure probabilities [7]. The *OPF* spoke solves AC optimal power flow on the base case using LUMINA [11], a foundation model for topology-transferable ACOPF that generalizes across grid topologies. The *contingency* spoke uses LUMINA's batch interface to evaluate outage scenarios in a single call, producing per-scenario severity scores from loading violations, voltage deviations, and power-balance error. These scores are used for screening only. GridMind maps the highest-ranked (lead time,

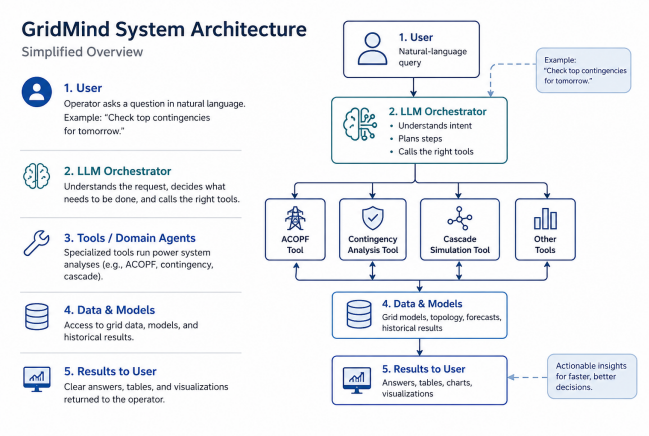

Figure 1: GridMind system architecture. The LLM orchestrator maps natural-language requests to domain-tool calls, while specialized backends perform grid analyses such as ACOPF, contingency screening, and cascade simulation. Persistent data and model artifacts provide the state needed to answer follow-up queries and reuse prior computational results.

ensemble member) pairs back to the original outage masks and passes those masks, together with the base case, to the *cascade* spoke. The cascade stage runs EPFast, a DC power-flow cascade simulator with threshold-based line tripping. External systems (AERIS and EPFast) are accessed through MCP-based integration, which isolate orchestration logic from transport details and allow backends to evolve without affecting the rest of the graph. The orchestrator summarizes the cascade results and presents key findings to the user in natural language.

3.3 Artifact-Based Orchestration

Grid and weather analysis computations produce large numerical data objects. A parsed transmission network can contain tens of thousands of buses and branches; an AC optimal power flow solution spans branch flow and voltage tables; weather access produces derived line-wind time series; cascade simulation and contingency analysis result in potentially thousands of scenarios. GridMind separates agent state from data: the agent holds only lightweight identifiers (run keys, status flags, and pointers to results), while all bulk data lives on disk in a persistent store called the *ArtifactStore*. Every artifact is written under a content-derived hash of its inputs, which we call its signature, so that any downstream stage or future session can locate the exact data product it needs without reconstructing the original computation. In our implementation, a signature is the SHA-256 digest of a canonical JSON serialization of the stage inputs: keys are sorted, defaults are materialized, and file inputs are represented by their own content fingerprints rather than mutable path names. We use signatures as artifact identities and orchestration targets, not as an adversarial security boundary; for the number of artifacts generated in our experiments, accidental collisions are negligible.

A naive agent would record which actions have run and use that log to determine what to do next. This fails when inputs change between turns, because reusing a stale intermediate artifact would silently corrupt downstream results. GridMind tracks progress through content hashes instead. Each stage's signature is a hash of its inputs: the network file determines the case signature; the OPF signature combines the case, solver, and outage schedule; the contingency signature adds the time window and scenario policy. A prerequisite is satisfied only when an artifact with the correct signature exists in the ArtifactStore. The orchestrator checks prerequisites in dependency order and executes the first unsatisfied one, giving the agent loop a well-defined termination condition. A change to any input invalidates only the affected stages and their descendants; unaffected stages are never rerun. The orchestrator also tracks whether a target signature has previously failed, distinguishing satisfied artifacts from failed ones to avoid looping on the same unsatisfied action. Query operations receive their own signatures derived from the artifact they read plus the query parameters, allowing multiple queries over one solution without rerunning the solver. This makes GridMind not only an execution pipeline but also an analysis interface over persisted artifacts, supporting summary views, ranked scenario lists, and critical-constraint queries.

3.4 Backend Contracts

Each spoke dispatches computation to a backend through a stable interface. The ACOPF spoke requires any solver to emit a standard set of branch, bus, and generator tables: LUMINA and classical solvers such as Pandapower [17] both satisfy this contract and are interchangeable from the orchestrator's perspective. Replacing a solver, forecast source, or simulator requires modifying only the corresponding spoke. In GridMind, LUMINA serves as a fast surrogate for ACOPF and post-contingency screening, but its solutions are not passed downstream. For base OPF, LUMINA returns bus voltages and angles, generator active and reactive power, line loadings and flows, power-balance error, and voltage violations, which GridMind converts into standardized tables. For contingency evaluation, LUMINA's batch interface returns a full per-scenario solution, but the contingency spoke reduces these to summary rows containing severity scores, maximum loading percentages, voltage violation counts, and power-balance errors. These diagnostics are used for ranking and inspection. The selected (lead time, ensemble member) pairs are then mapped back to the original outage masks from the outage schedule, and the cascade spoke receives these masks together with the base case file. It does not receive the LUMINA solution. EPFast re-solves independently under its own DC model, so cascade results do not depend on LUMINA's per-scenario AC solution accuracy.

4 Evaluation

4.1 Setup

We evaluate GridMind on the Texas 2000-bus synthetic network [3] (2,000 buses, 3,913 branches, 544 generators, 49,650 MW load). The weather scenario is AERIS-Laura: a synthetic Hurricane Laura simulation with 50 ensemble members over a 72-hour horizon. All five spokes are exercised against their live backends: AERIS for weather, the LOGON-based outage model for asset failures, LUMINA for

ACOPF and contingency screening, and EPFast for cascade simulation. The evaluation targets orchestration correctness (whether the right stages rerun when inputs change) and pipeline throughput.

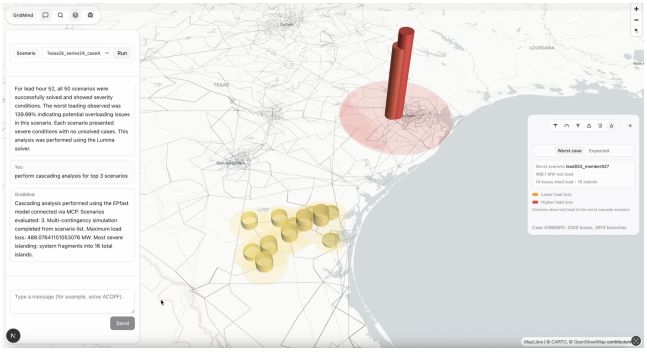

Figure 2: GridMind web interface after a cascading-outage analysis. The user first requests contingency screening and then asks for cascading analysis of the top-ranked scenarios; GridMind executes the EPFast backend and returns both a textual summary and a geospatial visualization of load loss and islanding on the Texas 2000-bus case.

4.2 Orchestration Behavior

We construct a seven-query session in which each follow-up changes one parameter and is expected to invalidate a different subset of the dependency graph. Table 1 shows the Hit/Run trace for each stage and query. The sequence covers five distinct invalidation patterns: leaf-only (k changes in Q2, Q3, Q6), side-branch (ACOPF solver swap in Q4), upstream cascade (forecast initialization shift in Q5), and full reuse (exact revert to Q1's configuration in Q7).

Here k denotes the number of ranked outage scenarios passed to the cascade simulator. Changing k should affect only the cascade artifact, because the weather, outage, ACOPF, and contingency artifacts are unchanged. Changing the ACOPF solver should affect only the ACOPF artifact in this configuration, because contingency screening consumes the outage schedule rather than the base OPF solution. By contrast, changing the forecast initialization changes the weather-derived line-wind artifact and therefore invalidates all downstream stages. The final query restores the exact Q1 configuration, testing whether GridMind can recover a previously generated artifact set rather than only reuse the immediately preceding state.

The H/R trace matches the dependency graph exactly on every query. Q2, Q3, and Q6 change only the cascade fan-out k, so only cascade re-executes. Q4 swaps the ACOPF solver: ACOPF re-executes, but contingency and cascade reuse because in this pipeline contingency consumes the outage schedule directly, not the ACOPF solution. Q5 shifts the forecast initialization by six hours, which changes the line-wind signature and propagates through all downstream stages; the orchestrator correctly recomputes all five. Q7 restores Q1's exact configuration and the entire pipeline returns in 0.02 s with no LUMINA or EPFast calls. Across the seven queries, GridMind issued 14 of 35 possible stage executions, a 60% reduction relative to full rerun. The end-to-end wall-time speedup is 2.12× (677 s vs. 1,433 s), bounded by the cascade calls that must run whenever k changes.

4.3 Pipeline Throughput

Table 2 shows per-stage cold-pipeline times for three distinct configurations exercised during the session. The cold pipeline completes in 170 s on the base configuration; contingency screening alone (LUMINA on 1,000 N-1 scenarios) accounts for 76.5% of cold time. ACOPF cost differs by an order of magnitude across solvers (LUMINA 1.8 s vs. pandapower 31.4 s). Cache-hit verification is free: Q7, which satisfies all five signatures, returns in 20 ms. Of the 756 s saved by reuse, about 585 s come from five cache-hit contingency screenings; the remainder comes from upstream reuse (line-wind, outages, ACOPF) and cascade reuse. This confirms that the heaviest single stage is also frequently reused across follow-up queries, which is the central design intent.

5 Discussion & Conclusion

GridMind shows that content-hash-based orchestration is a practical mechanism for interactive scientific agents that manage large, expensive data products. The evaluation makes three concrete points. First, the orchestrator routes a five-stage cross-domain pipeline correctly from a single natural-language request without explicit stage sequencing by the user. Second, partial reuse is automatic: across a seven-query session, GridMind executes 14 of 35 possible stages, while an exact revert to a prior configuration returns with no solver or simulator calls. Third, at the scale of the Texas 2000-bus network, reuse reduces end-to-end wall time by 2.12× relative to full rerun, making interactive follow-up analysis feasible within a session.

Current limitations include evaluation on a single hurricane scenario and grid case, the use of surrogate and reduced-order models for screening and cascade, and the absence of an eviction policy for the ArtifactStore. Operational use would require validation against utility-grade models, evaluation across hazard types, and storage lifecycle management.

Future directions include real SCADA/PMU integration and evaluation of the orchestration mechanism on other multi-domain scientific pipelines.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357.

References

[1] Carl S. Adorf, Paul M. Dodd, Vyas Ramasubramani, and Sharon C. Glotzer. 2018. Simple data and workflow management with the signac framework. *Computational Materials Science*. https://arxiv.org/abs/1611.03543.

[2] Emmanuel O. Badmus and Amritanshu Pandey. 2026. PowerDAG: reliable agentic AI system for automating distribution grid analysis. https://arxiv.org/abs/2603.17418.

[3] Adam B. Birchfield, Ti Xu, Kathleen M. Gegner, Komal S. Shetye, and Thomas J. Overbye. 2017. A synthetic texas footprint powergrid. *IEEE Transactions on Power Systems*, 32, 6, 4916–4926. doi:10.1109/TPWRS.2017.2698963.

[4] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. 2024. ChemCrow: augmenting large-language models with chemistry tools. *Nature Machine Intelligence*. https://arxiv.org/abs/2304.05376.

Table 1: Stage Hit/Run trace across the seven-query AERIS-backed session. H = content-hash signature already satisfied in the ArtifactStore; R = stage executed. Wall times are in seconds.

		line-wind	outages	ACOPF	contingency	cascade	Stages run	Total (s)
Q1	cold start, k=50	R 4.3	R 1.8	R 4.7	R 125.4	R 34.0	5/5	170.2
Q2	k=200 (cascade only)	H	H	H	H	R 86.4	1/5	86.4
Q3	k=500	H	H	H	H	R 199.7	1/5	199.8
Q4	ACOPF solver → pandapower	H	H	R 31.4	H	H	1/5	31.4
Q5	forecast init shifted +6 h	R 4.3	R 0.2	R 1.8	R 25.7	R 79.2	5/5	111.2
Q6	k=500 under shifted init	H	H	H	H	R 78.0	1/5	78.0
Q7	revert to Q1's exact configuration	H	H	H	H	H	0/5	0.02
						Reuse session cumulative	14/35	677.1
						Full-rerun ablation (each query in a fresh namespace)	35/35	1432.6

Table 2: Cold per-stage wall time (seconds) on the AERIS-backed Texas 2000-bus case for three configurations.

Stage	Q1	Q4	Q5
line-wind	4.24	4.13	4.11
outages	1.79	1.83	0.19
ACOPF	1.81	31.42	1.78
contingency	129.98	134.40	31.23
cascade	32.04	88.98	81.37
Total	169.86	260.76	118.68

[5] Quintina Campbell, Sam Cox, Jorge Medina, Brittany Watterson, and Andrew D. White. 2025. MDCrow: automating molecular dynamics workflows with large language models. https://arxiv.org/abs/2502.09565.

[6] Xu Chen, Fangning Ren, Dazhou Yu, Lechen Dong, and Fang Liu. 2025. AutoSolvate-Agent: an autonomous agent for GPU-accelerated solution-phase quantum chemistry. In *KDD 2025 Workshop on Large Language Models for Scientific and Societal Advances*. https://openreview.net/pdf/395d4c9eda83111a078ba05d86adcf84ec77e4aa.pdf.

[7] Ian Dobson, D. Adrian Maldonado, and Mihai Anitescu. 2026. The statistical spread of transmission outages on a fast protection time scale based on utility data. *IEEE Transactions on Power Systems*, 41, 1, (Jan. 2026), 296–306.

[8] Väinö Hatatpää et al. 2025. AERIS: argonne earth systems model for reliable and skillful predictions. https://arxiv.org/abs/2509.13523.

[9] [SW] LangChain AI, LangGraph: Building Stateful, Multi-Actor Applications with LLMs. URL: https://langchain-ai.github.io/langgraph.

[10] Simone Leo, Michael R. Crusoe, Laura Rodríguez-Navas, Raül Sirvent, Alexander Kanitz, et al. 2024. Recording provenance of workflow runs with RO-Crate. *PLOS ONE*. https://arxiv.org/abs/2312.07852.

[11] Yijiang Li, Zeeshan Memon, Hongwei Jin, et al. 2026. LUMINA: foundation models for topology transferable ACOPF. https://arxiv.org/abs/2603.04300.

[12] Felix Mölder et al. 2021. Sustainable data analysis with Snakemake. *F1000Research*, 10, 33. doi:10.12688/f1000research.29032.2.

[13] NOAA National Centers for Environmental Information. 2025. U.s. billion-dollar weather and climate disasters. (2025). doi:10.25921/stkw-7w73.

[14] Giovanni Pizzi, Andrea Cepellotti, Riccardo Sabatini, Nicola Marzari, and Boris Kozinsky. 2016. AiiDA: automated interactive infrastructure and database for computational science. *Computational Materials Science*. https://arxiv.org/abs/1504.01163.

[15] Edgar C. Portante, James A. Kavicky, Brian A. Craig, Leah E. Talaber, and Stephen M. Folga. 2017. Modeling electric power and natural gas system interdependencies. *Journal of Infrastructure Systems*, 23, 4, 04017035. eprint: https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29IS.1943-555X.0000395. doi:10.1061/(ASCE)IS.1943-555X.0000395.

[16] Junyi Shen, Noppanat Wadlom, Lingfeng Zhou, Dequan Wang, Xu Miao, Lei Fang, and Yao Lu. 2025. FlowMesh: a service fabric for composable LLM workflows. https://arxiv.org/abs/2510.26913.

[17] [SW] Leon Thurner et al., pandapower: An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems. URL: https://pandapower.readthedocs.io.

[18] Yihan Wen and Xin Chen. 2024. X-GridAgent: an LLM-powered agentic AI system for assisting power grid analysis. https://arxiv.org/abs/2512.20789.

[19] Yan Zhang, Ahmad Mohammad Saber, Amr Youssef, and Deepa Kundur. 2025. Grid-agent: an LLM-powered multi-agent system for power grid control. https://arxiv.org/abs/2508.05702.