Toward An Explainable Electric Power Grid Operation Assistant Using Large Language Models

by

Anish Ravichandran

B.S. Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2024

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2025

Authored by: Anish Ravichandran
Department of Electrical Engineering and Computer Science
January 17, 2025

Certified by: Marija D. Ilic
Joint Adjunct Professor, EECS, and
Senior Research Scientist, LIDS, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair
Master of Engineering Thesis Committee

Toward An Explainable Electric Power Grid Operation Assistant Using Large Language Models

by

Anish Ravichandran

Submitted to the Department of Electrical Engineering and Computer Science on January 17, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

This thesis explores potential applications of LLMs for assisting the analyses and decision-making of complex electric power grid operators. The power grid is a critical piece of infrastructure currently challenged by increased electrification, integration of renewable energy sources, and distributed energy resources (DERs). Human operators struggle to process the massive amounts of data produced by modern smart grids and need innovative solutions to handle the increased complexity of operational decisions. This thesis investigates the potential role of Large Language Models (LLMs) in grid operation tasks, focusing on interpretability and generalizability while exploring how LLMs can assist operators by providing actionable insights and recommendations. Multiple versions of LLM agents were developed, including naive and tool-assisted designs, and were evaluated on the Learn to Run a Power Network (L2RPN) benchmark for steady-state and cascading failure scenarios. While the LLM agents performed better in scenarios requiring exploratory decision-making, they struggled in steady-state operation and were constrained by their integration with tools and the testing environment. This work was limited by compute constraints, which affected the choice of model and the length of evaluation scenarios, and future work is needed toward seamless interaction of LLMs and power systems simulators, however LLMs have the potential to transform future grid operation, paving the way for more resilient and sustainable energy sector of the 21st century.

Thesis supervisor: Marija D. Ilic
Title: Joint Adjunct Professor, EECS, and
Senior Research Scientist, LIDS

Acknowledgments

First and foremost, I'd like to extend my deepest gratitude to Professor Ilić for her invaluable support and advice throughout my time at MIT. During my tumultuous RA journey with many twists and turns, she has been the beacon that guided me to shore. I'd also like to thank Guangchun (Grant) Ruan for his mentorship and practical advice during the thesis process, and the members of the Electrical Energy and Systems Group (EESG), who gave me the warmest welcome and always lent me an ear. I want to convey my sincere thanks to Guy Warner for his generous funding of my RAship this past year, which was instrumental in supporting my research and making my graduate studies possible. And lastly, I'm indebted to my family and friends. Without your support, I could not have come this far, and I am grateful for all of the lessons you have taught me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The power grid is an essential piece of infrastructure underlying nearly every aspect of modern society. From powering homes to enabling technological advancements, the grid serves as the backbone of contemporary life. However, in the 21st century, the grid is facing unprecedented challenges driven by the pressure of climate change and increasing electrification across many sectors, such as transportation and heating. These pressures create an urgent need to revamp the grid to meet rising demand while achieving carbon emissions targets. The transformation of the grid is full of obstacles, including increasing grid complexity, the integration of renewable energy sources, and distributed energy resources (DERs). Compounded with these challenges is the reluctance to build new infrastructure, forcing existing transmission and distribution networks, which were originally designed for steady power flows from fossil-fueled generators to centralized demand centers, to accommodate uncertain and highly variable power flows [1].

The field of grid operation is rapidly evolving and implementing emerging technologies, with a new push towards smarter grids and better data-driven approaches in energy systems. As the current regulatory landscape moves towards increased transparency, sensitivity, and privacy for the sake of increasing market participation and reducing energy prices, the sharing of open data and information is expected to significantly increase. While the availability of data can provide invaluable insights to improve grid operations, it also creates a bottleneck – the inability of human operators to analyze and act on massive amounts of incoming information effectively. To avoid overwhelming operators and compromising the reliability of the grid, there is a need for improved decision-making processes that can handle uncertainty and topological complexity in near-real time. Automating control rooms is an unavoidable solution to free operators from intensive tasks and allow them to focus on higher-level strategic decisions.

Artificial intelligence (AI) offers a potential solution to these challenges, making it possible to parse the large amounts of data and distill core insights and actionable recommendations for operators to consider. However, with a critical system like the power grid, human operators are not going to be replaced any time in the near future. Though AI systems may be able to analyze complex interactions within the grid and reduce cognitive load on operators, they must be able to be interpretable and trustable, especially if operators are to rely on the outputs of these models.

Large Language Models (LLMs) present an exciting opportunity for this domain. LLMs are AI models trained on an expansive corpus of text data that excel at communicating with humans in natural language, helping human decision-makers interface with raw data more effectively and efficiently. LLMs present a unique opportunity to create a grid operation assistant that can understand the grid operation task, interface with existing tools and other AI models to perform various analyses for the operator, and explain its reasoning in a manner that can garner the trust and confidence of human users.

This thesis explores the potential application of LLM agents to the grid operation task, examining their capability to make critical grid operation decisions in an interpretable and generalizable manner. By leveraging the unique capabilities of LLMs, this work aims to address real-world challenges that operators face today and demonstrate the potential of AI to transform how decisions are made in the energy sector, seeking to contribute to a more efficient, resilient, and sustainable power grid for the 21st century.

1.1 Background

1.1.1 Power System Trends

Smart Grids

A key innovation in power systems is the concept of the smart grid. Smart grids and conventional grids differ primarily through their communication and data flow. Smart grids incorporate Internet-of-Things (IoT) devices and real-time monitoring for quick fault detection. They are more flexible and adaptable to newer technologies, making it easier to incorporate renewable generation. Due to their enhanced communication protocols, smart grids open opportunities to advanced automation and remote control capabilities, along with increased consumer participation in the form of demand-response programs.

The increased data collected by grids are currently input into simulators such as PSS/E and MATPOWER, which provide environments to test algorithms and to train operators without causing any risk to the physical grid. One such simulator developed by Réseau de Transport d'Électricité (RTE), the French grid operator, is Grid2Op, which is leveraged in this work. Simulators provide a testbed for intelligent systems, and it is important that the complexity of the grid is modeled accurately for proper research.

With the increased flow of information expected to come from future industrial grids, even more advanced simulations can be created. Digital twins (DTs) are one such technology. DTs are virtual models of a physical grids system intended for scenario simulation, real-time diagnostics, and better optimization. A fully capable DT can provide real-time operational insights, identify weak spots in the grid for predictive maintenance, and test catastrophic scenarios in advance. As power grid DTs become more widely adopted among utilities and become tasked with more advanced decisions, there will be a need for increased interpretability and improved human-computer interaction to ensure alignment in goals [2].

Microgrids

Microgrids are localized power systems that are designed to be self-sustaining. They can be integrated with bulk power systems or operate in "islanded" mode, where they are cut off from the main grid and operate independently. There has been an increased interest in microgrid adoption over the past decade, and many projects have been attempted. However, microgrids experience frequent operational challenges, partly due to the complexity of building a system that can integrate with a main grid and maintain synchronization. Microgrids often have a higher penetration of renewable generation sources, causing them to be more vulnerable to power imbalances as generation fluctuates. Additionally, the nature of many microgrids require decentralized control, which raise cybersecurity concerns. To increase the success rate of microgrid projects, intelligent decision-making systems are required to ensure grid stability and increase reliability. Intelligent systems could assist in autonomous control, topology reconfiguration, and fault recovery, reducing the need to pay for excessive storage or dirty generators. These complex, decentralized systems have the most to gain through the use of AI operator assistants, especially for microgrids located on small islands, in remote areas, or without skilled operators.

DERs and Renewables

A major trend in power systems is the increase in distributed energy resources (DERs) and renewable energy penetration, due to global carbon emissions goals. Some examples of DERs are rooftop solar installations and battery storage systems in consumer buildings. With the popularity of policy incentives and decreased costs of solar PVs and batteries, the adoption of DERs has been steadily growing. However, the variability of solar production makes balancing supply and demand in real time more challenging, and DERs are less visible to grid operators, who have less control over them [3].

Renewables have also become popular in recent years, with a rising penetration of wind and solar power in global energy systems. Wind generation also experiences intermittency based on weather patterns, contributing to the challenge of integrating renewables. Also, large-scale solar production, such as that of solar farms, is often done in areas far from demand centers, requiring extra planning to integrate. High penetration of wind and solar in grids contribute to voltage and frequency instabilities and they necessitate flexible storage solution, driving up costs for grid planners. Grids with high penetration can also benefit from intelligent operator systems that could assist in real-time dispatch, fault prediction, and DER management. As the world pushes for carbon neutrality, the drive to adopt renewables will only increase. Decarbonization mandates cause an increase in grid complexity, making it more difficult for operators to maintain the reliability and resilience of grids.

1.1.2 Grid Operation

Grid Operators

Grid operators oversee real-time grid operation, with their responsibilities including balancing supply and demand in real time, managing grid stability through voltage and frequency control, and handling outages and ensuring rapid restoration of operation. The core system that grid operators use for real-time data monitoring and control is the Supervisory Control and Data Acquisition (SCADA) system, which has limited predictive capabilities and is highly dependent on operator input. Operators also use Energy Management Systems

(EMS), which offer centralized tools for optimization, contingency analysis, and economic dispatch. EMS often relies on optimization algorithms, but it is not very adaptable to highly dynamic scenarios like those involving DERs. Finally, operators have access to more intelligent contingency analysis tools, which assess potential failures and provide operators with ranked scenarios to mitigate risks. These tools are still highly manual, require significant onboarding to use, and are ill-prepared for newer grid scenarios involving high-penetration of renewables and DERs [4]. AI assistant systems could enhance operator tools by interpreting and responding to SCADA outputs autonomously, generating natural language recommendations for operators to factor into their decisions, and performing more explainable analyses in layman terms.

Decision Frameworks and Benchmarks

Structured decision-making is essential for a field as critical as power systems, and there are many formulations to solve grid operation problems such as calculating power flow, isolating faults, and minimizing load shedding. Most traditional frameworks rely on optimization, such as linear programming or nonlinear optimization methods, and are implemented with complex solvers that take a large number of parameters. For complex grid scenarios, these methods can accrue high computational costs, and many of the frameworks used in industry still struggle with unpredictable events like DER fluctuations.

To evaluate the effectiveness of proposed methods to solve these grid operation problems, groups like the Institute of Electrical and Electronics Engineers (IEEE) have developed test systems that are widely used in the literature. In this work, the Grid2Op system uses IEEE test systems as a base for its dynamic environment simulations, which serve as a more recent benchmark to evaluate agent performance in grid operation scenarios. These scenarios evaluate robustness to failures, adaptability to renewable integration, and trustworthiness of agents in near-failure scenarios. This work builds on Grid2Op by using it as a benchmark to evaluate large language model agents, and it helps demonstrate the adaptability and decision-making abilities of these agents in dynamic and realistic grid scenarios, especially when used in conjunction with traditional frameworks.

Operation Challenges

A main challenge in grid operation currently is the aging of grid infrastructure. Many grids rely on outdated equipment, which have an increased risk of failure and are ill-equipped to handle modern loads and renewable generation. Another challenge is the high cost of operation of the grid, where equipment maintenance costs are increasing. There is a constant pressure on operators to reduce costs while improving efficiency. Finally, existing tools are often rule-based and unable to adapt to rapidly changing conditions.

As the grid becomes more digitized, it will become more vulnerable to cyberattacks. Future grid systems could employ AI to identify and mitigate cybersecurity risks by analyzing threat patterns and responding without the need of a human. Another challenge emerging with the changing climate is the increasing frequency of extreme weather events such as storms and heatwaves. This will make maintaining grid resilience a larger challenge, requiring tools that can better forecast and adapt to weather-related events. Additionally, as more DERs are integrated into the grid, centralized control will become less viable. Current systems struggle with decentralized energy sources and will need to adopt new approaches to the grid operation problem. Intelligent grid operation agents offer adaptability to dynamic conditions, natural language explanations and actionable insights for operators, reducing the turnaround time for decisions. They also potentially provide scalability in the form of multi-agent systems that can handle distributed grid components. This work aims to highlight the capability of these agents to address both current and emerging challenges.

1.1.3 Large Language Models

History and Development

Large Language Models (LLMs) are the latest iteration of a family of models coming from Natural Language Processing (NLP). Early NLP models were simple statistical models, such as N-grams or Bag-of-Words (BoW) models, which would compute the next word based on simple pair-wise probabilities. In the past two decades, NLP has transitioned to neural network-based models like Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and most recently attention mechanisms. In Vaswani et al.'s Attention Is All You Need (2017), the self-attention mechanism was applied for contextual understanding in the novel transformer architecture, and it revolutionized NLP by enabling parallel processing and long-range dependencies between inputs. Shortly after, the transformer architecture was scaled up by industry actors like Google and OpenAI into larger models trained on larger corpora of data, leading to the LLMs seen today. From the initial GPT-1 with 117M parameters to the large-scale models like GPT-4 [5] with hundreds of billions of parameters, increasing the number of parameters caused the models to exhibit emergent abilities, such as instruction following, zero-shot reasoning, and few-shot learning. Because the grid operation task requires processing complex, interconnected data, the transformer's ability to model sequential and contextual relationships can be effectively leveraged. LLMs are versatile and can perform diverse tasks without needing to be re-engineered, making them a great fit for power grid applications.

Prompt Engineering and Finetuning

Though LLMs are powerful tools that can be applied to a variety of tasks, their behaviors must be guided to be effective in real-life scenarios. The field of prompt engineering centers around crafting effective inputs to guide LLM behaviors. Some examples of prompt engineering are few-shot prompts, which provide a limited set of examples for the LLM to learn from and emulate during its response, and Chain-of-Thought (CoT), which outlines a step-by-step reasoning framework for the LLM to follow. Prompt-only methods have limitations, however, with the results being highly dependent on the quality of the prompts, and in unstructured scenarios the LLM still has a high risk of errors.

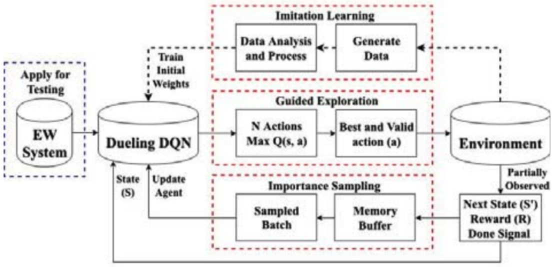
The field of fine-tuning also aims to adapt pre-trained models to specific tasks or domains, but through the process of retraining the model on a domain-specific examples. Supervised fine-tuning incorporates labeled datasets, while RL fine-tuning optimizes model behavior based on feedback from an environment. Most LLMs used in industry are fine-tuned with one or both of these methods to help specialize them to more technical domains and align the model outputs to the constraints of the domain. In this work, only prompt engineering is employed due to cost constraints, but designers of the AI grid operation assistants of the future will likely employ both prompt engineering and fine-tuning to refine the performance of their models.

LLM-Based Agent Systems

There is a growing trend of integrating LLMs in autonomous and semi-autonomous decision-making systems, and several frameworks have emerged in the literature. The Reasoning and Acting (ReAct) framework [6] combines the LLM agent's ability to think in steps with its need to interact with the environment. LLMs are able to dynamically generate plans of action and execute them iteratively. This framework is suited for lower-level tasks that require quick action-taking and limited reflection.

The tool-use framework enables LLMs to interact with external APIs, simulators, or databases to acquire knowledge outside the training set of the LLM or perform tasks that are ill-suited for language models. For example, if a user wants to know how much hotter the temperature is today than the monthly average, the LLM would call the weather API and fetch the daily temperatures for the past month, then call a calculator tool and supply it with this temperature data to calculate the average and subtract this average from today's temperature. The LLM would then contextualize the calculator output in natural language for the user. In the power systems context, LLMs can query SCADA systems for current grid state information and use simulator tools like Grid2Op to test potential grid operations before deployment. The integration of tools enhances LLM's real-time decision-making capabilities. The ability of LLMs to generate code is an emergent property found in larger models that have been trained or fine-tuned on large corpora of code. For tasks where the LLM needs to solve dynamic problems or implement custom strategies, it can generate domain-specific scripts on the fly.

For more complex problem-solving tasks, the Chain-of-Thought (CoT) reasoning framework was established, where LLMs are prompted to perform multi-step reasoning on the data within its context window. This improves transparency by providing interpretable reasoning paths for human users to follow. For distributed tasks that may be too complex for a single LLM agent, the Mixture-of-Expert (MoE) framework outlines how multiple agents can be organized to focus on specific sub-tasks. For example in the power systems context, the first agent can monitor DER performance, the second agent can optimize grid configuration in real time, while the third agent can generate actionable insights for operators. The advantage of this approach is that the type of agent can be optimized for the specific subtask, and non-LLM agents can be integrated if they suit the task better. In this work, ReAct and CoT align with the dynamic and sequential nature of grid operation decisions in Grid2Op, while tool use allows the LLM to integrate with simulators and base its chosen actions in the physics of the grid.



(a) Team GEIRINA's Agent Architecture

TABLE I. PERFORMANCE COMPARISON OF DIFFERENT AGENTS ON 200 UNSEEN SCENARIOS WITH 288 TIME STEPS

Agent	Game Over	Mean Score All	Mean Score w/o Dead
Do Nothing	91	2471.42	**4534.72**
Only Imitation	198	38.21	3820.63
Guided Trained	7	4269.63	4424.49
EW $\lambda = 0.85$	**0**	4253.40	4253.40
EW $\lambda = 0.875$	1	4347.56	4369.41
EW $\lambda = 0.90$	**0**	4396.77	4396.77
EW $\lambda = 0.925$	**0**	**4493.27**	**4493.27**
EW $\lambda = 0.95$	**0**	4492.89	4492.89
EW $\lambda = 0.975$	2	4446.12	4491.03

(b) Team GEIRINA's Training Scores

Figure 1.1: Team GEIRINA's Architecture and Training Scores [8]

1.2 Related Work

1.2.1 AI Agents for Grid Operation

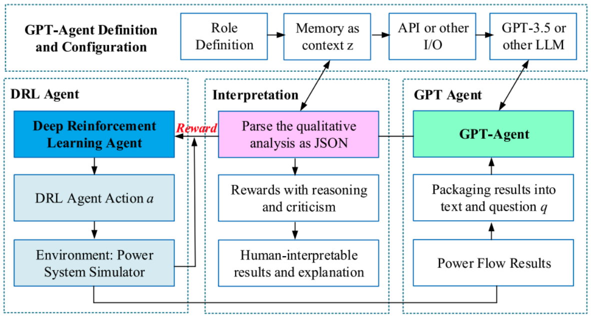
In recent years there have been attempts to define power grid operation benchmarks for agents. The Learn to Run a Power Network (L2RPN) challenge [7], developed by RTE and the Electric Power Research Institute (EPRI), features a grid operation simulation platform, Grid2Op, that aimed to standardize agent interactions with the power grid and simulate their outcomes. The challenge hosted competitions at various conferences from 2019 to 2023, and the winners of the competitions have open-sourced their code for benchmarking purposes. All of the winners employed deep reinforcement learning (DRL) methods to address the challenges in power grid optimization, such as the non-convex nature of AC power flow with security constraints and the exponential growth of the action space when more lines and busbars were introduced into the grid.

IJCNN 2019 Winner

The winner of the IJCNN 2019 competition, the AI and System Analytics team at GEIRI North America, used a combination of of imitation learning (IL) to provide a an initial policy for the AI agent, then using dueling deep Q-learning (DDQN) methods with a novel guided exploration technique to improve efficiency during training [8]. For longer testing periods, the team experimented with an early warning mechanism tuned to different line loading thresholds that helped the agent find better topology control strategies. Ultimately, the team was successful because they were able to downsample the action space into a more manageable problem for their RL agent, which led to more effective training.

NeurIPS 2020 Winner

The NeurIPS 2020 L2RPN challenge featured a larger grid with a more challenging set of competition features, like maintenance events, opponents attacking the grid, and more difficult generation and load profiles. The winning team for this competition was from Baidu, which proposed a novel search-based planning method called Search with the Action Set (SAS) [9], which would sample the top-K actions for a given timestep from a



(a) Architecture of Hybrid LLM-DRL Agent

Method	Average generation costs (USD$)	Average performance score evaluated by GPT-Agent (linguistic reward)	Average contingency constraints violation (%)	Qualitative objectives
OPF based on IP method for optimization (benchmark)	1.2393e5	80.87	0.0404	No
Supervised learning	1.2532e5	39.58	0.6963	No
Proposed method	**1.2575e5**	**94.50**	**0.0000**	**Yes**
Safe DRL without GPT-agent	1.2540e5	85.63	0.0080	No

(b) Performance of GPT-Agent in OPF Scenario

Figure 1.2: Architecture and Performance of GPT-Agent in OPF with Linguistic Stipulations [10]

black-box policy network and run a simulation function over these actions, choosing the action that followed the environmental constraint of the grid and returned the highest anticipated reward. During training, the team was able to improve a given black-box policy network on these environmental constraints, leading to better actions selected by the agent during the competition. This method led the team to win both tracks of the NeurIPS 2020 competition, because their agent was able to thrive in the face of adversarial attacks, with multi-energy mixed grids, and across different topologies.

Both of these winners' approaches utilized black-box networks trained on the specific topology that they were tested on. The current literature does not cover topology-agnostic and more interpretable AI models for the grid operation task, which could be a potential contribution of large language model agents.

1.2.2 Large Language Models in Power Systems

There has been limited application of large language models in power systems, with the most notable work being released in the past year. Reference [10] tackles the problem of aligning existing RL agents with the constraints of grid operation manuals. The agent used OpenAI's GPT-3.5 as the LLM, which would run a grid scenario with a power system simulator. The power flow results were then fed into the LLM, which would parse the power flow results in JavaScript Object Notation (JSON) format and decide on a reward to give the DRL agent, which would then choose its next action. The LLM would also have access to a memory module, which contained an operation manual for the IEEE 118-bus grid with linguistic stipulations, and the recommendations from this manual were used by the LLM to shape the reward function for the RL agent. The results in figure 1.2 shows how the hybrid LLM-DRL agent was successfully able to reduce grid constraint violations and maintain comparable generation costs while being able to handle natural language objectives.

Reference [11] outlines the development of foundation models for the electric power grid, or GridFMs, which are inspired by purely text-based LLMs and plan to use the same training process on the graph neural network (GNN) architecture to address critical power systems problems like load flow, system security, and electricity markets analysis. Future versions plan to fully integrate language models and models trained on other modes of data for more sophisticated analysis. The advantage of the proposed GridFM would be the speed-up in computation compared to existing solvers, adaptability to different topologies, and ability to enable decentralized grid control. There are significant hurdles to accomplishing this goal, as a large portion of the potential training data is siloed within heterogeneous entities, and there needs to be more policy initiatives to encourage open-sourcing data for such models. While these domain-specific foundation models are in development, it is important to test the capabilities of existing models. Due to their extensive training sets, LLMs not only have language generation capacity, but some level of power systems knowledge, though there are no benchmarks in the literature to quantify this domain expertise. This work aims to answer this question, further explore the most effective use-case of LLMs in power systems, and explain potential limitations users may experience in application.

Chapter 2

Methodology

2.1 Grid Simulation and Benchmarks

The grid operation task covers a wide variety of subtasks, ranging from preventing cascading faults on the millisecond level to designing equipment maintenance schedules on the order of years. Therefore it is necessary to rescope the task to specific subtasks that can be quantified and evaluated against ground truth data for this thesis. For the L2RPN challenge, the Grid2Op framework was developed to help narrow down the problem to topology optimization.

2.1.1 Grid2Op

Grid2Op [12] is a Python module developed by RTE to facilitate experimentation on sequential decision making tasks for the power grid. Internally, the sequential decision making problem is modeled as a Markov Decision Process (MDP), which is very common in reinforcement learning (RL) contexts.

Markov Decision Processes

In an MDP the agent or policy takes some action $a_t \in \mathcal{A}$. This action is then processed by the environment, which updates its internal state from $s_t \in \mathcal{S}$ to $s_{t+1} \in \mathcal{S}$ and computes a reward $r_{t+1} \in [0, 1]$. This state and reward, (s_t, r_t), is then given to the agent or policy which in turn produces the subsequent action a_{t+1}. The list $s_1 \rightarrow a_1 \rightarrow (s_2, r_2) \rightarrow \cdots \rightarrow a_{T-1} \rightarrow (s_T, r_T)$ is called an "episode", composed of T "steps". Formally an MDP is defined by the state space, $\mathcal{S}$, the action space, $\mathcal{A}$, the probability distribution (over $\mathcal{S}$) that gives the next state after taking action a in state s, $\mathcal{L}_s(s, a)$, the probability distribution (over $[0, 1]$) that gives the reward r after taking action a in state s which lead to state s', $\mathcal{L}_r(s, s', a)$, and the maximum number of steps for an episode, $T \in N^*$ [12].

The main goal of a finite horizon MDP is to find a policy $\pi \in \Pi$ that, given states s and reward r, outputs an action a such that:

$$
\begin{aligned}
&\min_{\pi \in \Pi} \sum_{t=1}^{T} E\left(r_t\right) \\
&\text{s.t.} \\
&\quad a_t \sim \pi(s_t) \forall t && \text{policy produces the action} \\
&\quad s_{t+1} \sim \mathcal{L}_S(s_t, a_t) \forall t && \text{environment produces next state} \\
&\quad r_{t+1} \sim \mathcal{L}_r(s_t, a_t, s_{t+1}) \forall t && \text{environment produces next reward}
\end{aligned}
$$

Grid2Op Implementation

To redefine the MDP in Grid2Op, we need to define a simulator and time series input. The simulator is represented in Grid2Op's backend, and computes portions of the state space $\mathcal{S}$ such as the flows on powerlines, the active production value of generators, and the demand values of the loads. Therefore the simulator is essential to the computation of the transition kernel $\mathcal{L}_s(s, a)$. We can model this simulator with a function Sim that takes an input space $\mathcal{S}_{\text{im}}^{(\text{in})}$ and results in data in the output space $\mathcal{S}_{\text{im}}^{(\text{out})}$.

The final type of data to define the MDP is time series, which define what each consumer would consume and what each producer would produce if they were connected together with without any powerline constraints. Grid2Op assumes time series are balanced, meaning that producers produce exactly the right amount for the consumer to consume at each time step. These time series will be defined as input $\mathcal{X}_t$, representing generator active production (in MW), load active power consumption (in MW), load reactive consumption (in MVAr), and generator voltage setpoints (in kV).

For the purposes of the benchmark, we assume the structure of the grid will not change during a given episode. We can define the following properties: n_{line} is the number of powerlines or transformers in the grid, n_{gen} is the number of generators, and n_{load} is the number of consumers, usually representing a town or industrial site. For the scope of this thesis the scenarios with storage were not considered. All of these elements are connected together at substations, of which there are n_{sub} substations in the grid. Each substation can be divided into $n_{busbar\ per\ sub}$, which will be 2 for the purposes of the thesis, meaning each substation can have two separate electrical nodes at maximum. The environment representation of a grid provides a variety of other information that may be useful to the agent, such as whether a generator represents a renewable intermittent source, whether a generator can be redispatched by an agent, the minimum and maximum ramp rates, and much more.

2.1.2 L2RPN Rules

The objective of the L2RPN challenge is to successfully manage to operate a power grid, meaning the agent can change the topology or modify the productions to make sure it remains safe while being operated to minimize the energy losses. The simulation is turn-based as described in the previous section, so the agent will observe the environment, submit an action, and wait for the next observation. The agent will immediately fail if the power grid isn't operating properly, defined by the following two conditions:

1. Consumption is not met because no electricity is flowing to some loads or more than one power plant gets disconnected.

2. The grid gets split apart into isolated sub-grids making the whole grid non-convex.

These conditions can appear when power lines in the grid get disconnected after being overloaded. When a line get disconnected, it loads gets distributed over other power lines, which in turn might get overloaded and thus disconnected as well, leading to a cascading failure. When the power in a line increases above its thermal limit, the line becomes overloaded. It can stay overloaded for few time steps before it gets disconnected, if no proper action is taken to relieve this overload within 2 time steps, or 10 minutes of real time, this is a soft overload. If the overload is too high, above 200% of the thermal limit, the line gets disconnected immediately. This is a hard overload. At some point this can lead to a very rapid cascading failure in a single time step if some lines already got disconnected and other lines get quite loaded.

At each time step, the agent will be given current productions, loads, and more importantly the flows over the lines and the topology of the grid. It will then have to choose to perform zero, one, or multiple actions. Actions can consist of disconnecting or reconnecting a powerline, changing the topology of the grid (choosing to isolate some objects like productions, loads, powerlines from others), modifying the production set point of nonrenewable generators with redispatching actions, curtailing renewable generators to reduce their production. Due to this thesis being scoped to topological optimization, redispatching and curtailment weren't included in the implementation. Some actions can be considered illegal by Grid2Op if they don't comply with constraints (e.g. redispatching a generator above its maximum generation capacity). In that case, no action will be taken at that time step, similar to a do-nothing action [13].

2.1.3 Scoring

The scoring for the L2RPN challenge is based on minimizing energy losses. Transporting electricity always generates energy losses $\mathcal{E}_{loss}(t)$ due to the Joule effect in resistive power lines. At any time t:

$$\mathcal{E}_{loss}(t) = \sum_{l=1}^{n_l} r_l * y_l(t)^2 \tag{2.1}$$

Where r_l is the power line's resistance and y_l is the current flowing through it. At any time t, the grid operator is responsible for compensating energy losses by purchasing an equal amount of production at the marginal price $p(t)$ set by the energy market. Therefore:

$$c_{loss}(t) = \mathcal{E}_{loss}(t) * p(t) \tag{2.2}$$

Decisions made by the operator can induce costs as well, because when they require market players to perform specific actions, those producers should be compensated. Topological actions have no cost since the grid belongs to the operator, however energy producers are affected by actions that redispatch generation and must be paid. When an action redispatches some energy $\mathcal{E}_{redispatch}(t)$, some power plants will increase production by $\mathcal{E}_{redispatch}(t)$ while others will compensate by decreasing production by the same amount to keep the grid balanced. The grid operator will pay both producers at an additional cost $c_{redispatching(t)}$, which is higher than the marginal price $p(t)$ by some factor α:

$$c_{redispatching}(t) = 2 * \mathcal{E}_{redispatch}(t) * \alpha p(t),\ \ \alpha 1 \tag{2.3}$$

The first producer is paid an extra $\mathcal{E}_{redispatch}(t) * \alpha p(t)$ because they has to produce $\mathcal{E}_{redispatch}(t)$ more energy than planned, and the second producer also is paid an extra $\mathcal{E}_{redispatch} * \alpha p(t)$ to compensate for the $\mathcal{E}_{redispatch}(t)$ energy that it did not sell. The curtailment actions behave similarly to redispatching actions since they require market players to increase or reduce their production. Therefore, $c_{curtailment}(t)$ has the same formula as $c_{redispatching}(t)$:

$$c_{curtailment}(t) = 2 * \mathcal{E}_{curtailment}(t) * \alpha p(t),\ \ \alpha \geq 1 \tag{2.4}$$

Therefore our overall operational cost $c_{\text{operations}}(t)$ is

$$c_{\text{operations}}(t) = c_{\text{loss}}(t) + c_{\text{redispatching}}(t) + c_{\text{curtailment}}(t) \tag{2.5}$$

An agent can either manage to operate the grid for the entire scenario ($t_{\text{end}} = T_e$) or fail after some time t_{end} due to a blackout, in which case the cost $c_{\text{blackout}}(t)$ at a given time t would be proportional to the amount of consumption that was not supplied, Load(t), at a price higher than the marginal price $p(t)$ by some factor β:

$$c_{\text{blackout}}(t) = \text{Load}(t) * \beta * p(t),\ \ \beta \geq 1 \tag{2.6}$$

The cost of a blackout is a lot higher than the cost of operating the grid as expected. In real life, a blackout does not last forever and power grids restart, but this complexity is not considered by Grid2Op, so the scenario terminates. Therefore the total cost c for an episode:

$$Raw\_Score_{OperationCost}(e) = \sum_{t=1}^{t_{\text{end}}} c_{\text{operations}}(t) + \sum_{t=t_{\text{end}}}^{T_e} c_{\text{blackout}}(t) \tag{2.7}$$

Agents are encouraged to operate the grid for as long as possible, and will be penalized for a blackout even after the game is over, until T_e. In practice, over the course of an episode, the scores of agents can reach the billions, making it difficult to read. Therefore, during the competition a linear transformation was applied such that the score is 100 for an agent that handles all the scenarios, without using redispatching actions, with minimal losses of 1% for all the scenarios, and the score is 0 for the ‘DoNothing‘ agent [13]. However for this thesis, since the episode lengths were only one hour, this linear transformation was not necessary.

2.1.4 Environments

Due to Grid2Op being modeled after OpenAI Gym, the grid simulators are called environments. For benchmarking, the LLM agents will be evaluated on the IJCNN 2019 competition benchmark grid using the L2RPN 2020 reward function to calculate score. This

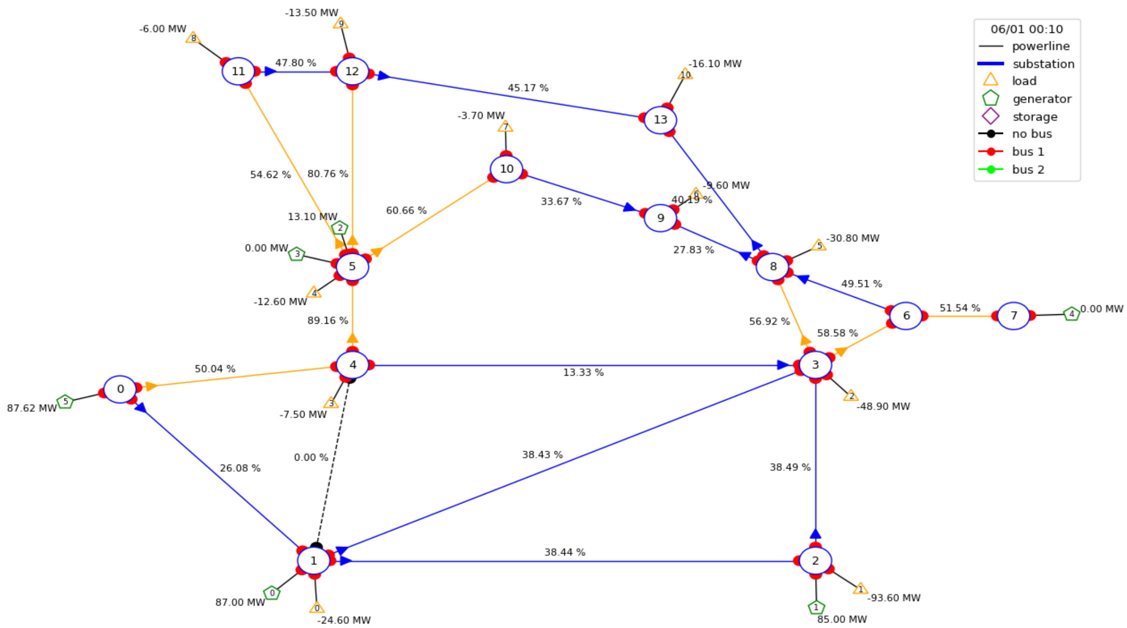


Figure 2.1: IJCNN 2019 Grid Configuration [13]

dataset is based on the IEEE Case 14 power grid, though it is slightly modified for the competition with a few generators added. It has 14 substations, 20 lines, 6 generators and 11 loads. This dataset was used by contestants during the warm-up phase of the competition for testing their initial implementations. It is a relatively small grid with no maintenance events built in. The time series data comes from one month of operation, with generation and load profiles at 5-minute resolution. For this competition, only topology changing actions were permitted.

In this environment one-hour-long scenarios will be used for benchmarking due to compute limitations. The first scenario is during steady state operation, where no lines are over loaded and the power flows are stable, and the second scenario is when lines 12 and 15, the lines connecting substations 9 and 10 and 3 and 6, respectively, are forcibly disconnected upon initialization, causing surrounding lines to become overloaded and eventually causing cascading failures if left unattended. See Appendix B for a detailed run of the scenarios.

2.1.5 Baselines

For the L2RPN environments, baselines take the form of agents that perform actions dictated by their corresponding method. These baseline agents are included as part of the Grid2Op framework or in the `L2RPN_Baselines` package.

DoNothing and Random Agents

The DoNothing Agent [12] always submits an empty action during when presented with an environment observation, while the Random Agent [12] will sample an action from the environment's action space at random and submit. Though the DoNothing Agent may seem like a poor baseline, it is worth noting that Grid2Op environments differ from other reinforcement learning environments due to the fact that by default the grid simulation starts at steady state and does not become unstable unless acted upon by an agent or another external actor.

Simulator-Based Agents

Grid2Op also provides some baseline agents that rely on open-source simulators. Expert Agent, designed by RTE, uses ExpertOp4Grid [14] in its backend, which is an expert system that uses a greedy algorithm to address overloaded lines as they occur. Expert Agent generates cheap but non-linear topological actions to solve overloads, and requires no training before being deployed. The ExpertOp4Grid simulator computes an influence graph around the overload of interest and ranks various substation busbar topologies based on their performance in simulation over the next timestep. The performance is assigned a score from 0 to 4, where 4 means the new topology solves all overloads, 3 means the topology solves only the overload of interest, 2 means the topology partially solves the overload of interest without causing new overloads or worsening existing ones, 1 means the topology solves the overload of interest but worsens other overloads, and 0 means the topology causes the agent to fail the Grid2Op task.

2.2 LLM Agent Design

2.2.1 Model Choice

The model chosen for this task was the closed-source `gpt-4-0125-preview` model by OpenAI, and prompts and answers were streamed via OpenAI's developer API. GPT-4 was chosen due to its parameters size, which allowed it to perform advanced reasoning on the large action space for this problem and utilize power systems knowledge from its expansive training set. Other smaller models, such as `Zephyr-7b-Instruct` (a fine-tuned variant of Meta's LLaMA) and `Mistral-7B-Instruct-v0.3` were explored, but these models experienced cognitive overload from the size of the grid topology prompt, suggesting that a larger number of parameters or more advanced prompt engineering is required to perform the grid operations task.

2.2.2 ReAct Framework

The initial implementation of the LLM agent utilized the ReAct framework across the whole episode, where Grid2Op would generate an observation from the power system simulator backend, then feed the observation for the current timestep to the agent. The LLM would parse this grid observation, perform unaided reasoning on the topology and possible actions, and return an action in Grid2Op's specific format to the environment. The environment would then input that action into the backend simulator, recalculate power flows, and present the observation for the next timestep. This implementation of the agent used LangChain's `ChatOpenAI` module to wrap around the GPT-4 LLM, and Pydantic schemas were bound to the prompts and output parsers to handle the structured JSON outputs. Each prompt to the

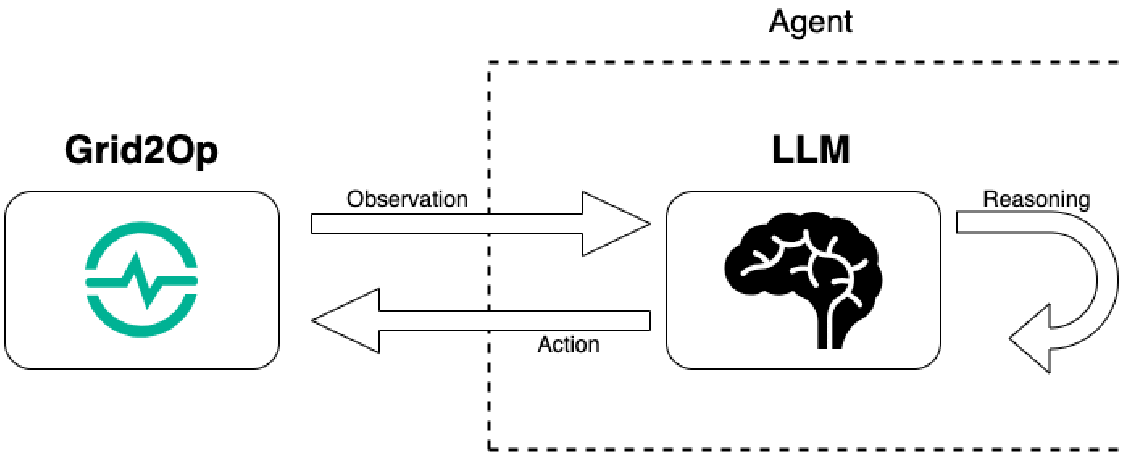


Figure 2.2: Naive ReAct Implementation

LLM would be stored in a message queue, and the LLM would be able to access the entire queue of messages across timesteps when deciding actions.

Many iterations of the prompting were made due to the complex formatting of the actions that needed to be output. Without providing at least a few schema examples in the initial prompt, the LLM would hallucinate its own schema, often resorting to JSON dictionaries. Additionally, when prompted to only return an action, the LLM would always return a "do-nothing" action, but when prompted to reason and adding a rationale section to the output schema, the LLM would begin to output actions. Perhaps due to OpenAI's post-training procedures, the LLM often hesitated to provide concrete actions, often opting for generic advice or stating that it would be best to use a simulator, but after adjusting the prompting to reflect the game-like nature of the scenario and reinforcing that the actions were not being utilized on a real power grid, the rate of nonresponses decreased.

2.2.3 Nested ReAct Framework with Tool Use

For the more advanced implementation, a nested ReAct framework was utilized. Similar to the naive implementation, this version also would perform reasoning on the observation returned by the environment, and in turn would compute an action, therefore using the ReAct framework across timesteps. However, within a single timestep the ReAct framework would be used an additional time to allow the LLM to engage with the tools provided and reason on their outputs before deciding on a single action. This agent was implemented in LangGraph, a library built on LangChain that allows agents to behave according to a behavior graph. Each tool call or reasoning step is represented by a node in the graph, and the LLM decides the next node to transition to in the edges of the graph. The same ChatOpenAI wrapper was used, and each tool had its own Pydantic schema due to the different input requirements for the simulators.

Four main tools were available to the LLM, with two additional tools designed to enhance the LLM's capabilities. The `RankOverloads` tool would take the Grid2Op observation and return a JSON dictionary encoding a ranking of the different overloaded lines by their

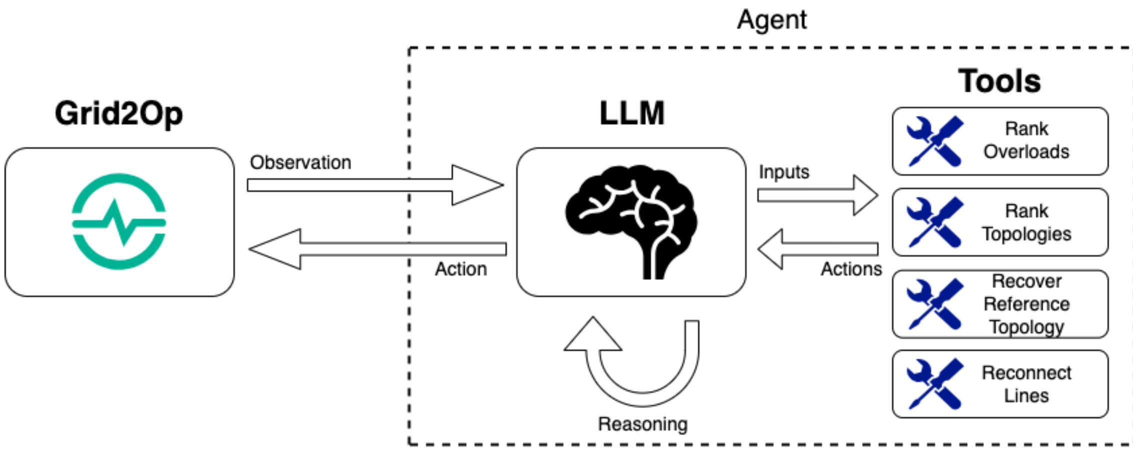


Figure 2.3: Nested ReAct Framework with Tool Use

criticality, which is the number of timesteps before the line gets disconnected, then by their loading percentage. The `RankTopologies` tool would run ExpertOp4Grid's simulator for an overloaded line of the LLM's choosing, returning the action that corresponds to the highest-ranking topology solution. The `RecoverReferenceTopology` tool would take a substation ID as input and attempt to generate an action that reverted the busbar configurations to the original topology during steady-state operation. The `ReconnectLines` tool would take a line of the LLM's choosing and attempt to generate an action that reconnected this line. The additional two tools consisted of the `CombineActions` tool, which would take two or more previously generated actions and attempt to combine them into a new legal Grid2Op action, and the `SimulateAction` tool, which would simulate the effect of an action taken on the current observation and return the output of the simulation, namely the resulting line loadings. These tools were designed to prevent the LLM from synthesizing actions from scratch and to let it focus on higher-level reasoning.

This implementation required many iterations of the prompting as well, where to guarantee that the intended inputs to tools were being parsed correctly, the LLM would first be prompted to choose a tool, then would subsequently be prompted to provide the specific input parameters for that tool. During testing over many timesteps, it became clear that GPT-4's context window became saturated by the queue of messages, because there were many more messages in this implementation from the tool choice prompts, the tool input parameter prompts, and the tool output messages. A decision was made to have the LLM summarize all of the messages at the end of a timestep, spelling out its interpretation of the grid observation, its goals, the tools it chose, and ultimately why it decided on the submitted action. The message queue would be cleared and replaced with this summary message and the original task prompt to maintain the LLM's train of thought over subsequent timesteps.

Chapter 3

Results

The objective of this thesis was to explore the application of large language model agents in grid operation contexts. The results of the agent implementations offer insights into the challenges and limitations of applying LLMs to power grid operations, particularly within the topology reconfiguration subtask of the L2RPN challenge benchmark.

3.1 Observations

On a qualitative level, both implementations of the LLM agent demonstrate that LLMs understand the grid operation task's rules and can remember the goal of minimizing economic losses. The agents demonstrate enough domain knowledge to describe actions on a high level, and they even identify their own limitations.

Naive ReAct Agent

For example, during steady-state operation scenarios in for the IEEE Case 14 environment, here is an excerpt of the naive LLM agent response:

```
Based on the provided grid state, it appears that all lines are
   currently operational (status: True) and there are no immediate
   overloads (all loading values are below 100%). However, some lines
    are approaching their capacity limits, notably:

- Line 7 (5_10_7) with a loading of 66.669387%
- Line 9 (5_12_9) with a loading of 79.88288%

Given the current state, there's no immediate action required to
   prevent overloads or disconnections. However, to proactively
   manage the grid and minimize the risk of future overloads,
   especially considering the dynamic nature of load and generation,
   it might be beneficial to adjust the topology to balance the load
   more evenly across the network.
```

```
Since the simulation is turn-based and considering the current state
   does not demand urgent action, the best approach is to monitor the
    grid closely in the next timestep for any significant changes in
   line loading, especially for lines nearing their capacity. This
   strategy allows for a more informed decision-making process,
   leveraging real-time data to optimize grid operations effectively.

Therefore, the recommended action for this timestep is to take no
   immediate action but to prepare for potential topology adjustments
    in the near future if line loadings approach critical levels.
   Current line loadings are below critical levels, and all lines are
    operational. No immediate action is required. Monitoring and
   preparing for potential future topology adjustments to balance the
    load more evenly across the network is advised.
```

The GPT-4 LLM demonstrates it can parse the grid topology from the input observation and understand where the line loadings are. It also describes its reasoning for why it chose a "do-nothing" action and described how this is subject to change in future timesteps. This clear reasoning can help build trust when interfacing with a human user. In a later timestep, when the Grid2Op environment adjusts generation and load values and some lines are attacked, the naive LLM agent presents its analysis:

```
Given the current state of the grid, the most critical observation is
    the loading on the lines, especially those approaching or
   exceeding their thermal limits, which could lead to disconnections
    if not addressed. The line with the ID 9 ("5_12_9") has a loading
    of 0.8760237 (87.60237%), which is significantly high and could
   risk disconnection if the load increases further. This line
   connects substation 5 to substation 12, indicating it's a crucial
   connection likely supporting a significant portion of the grid's
   power flow.

One approach to alleviate the load on a heavily loaded line is to
   change the topology at substations connected by the line. By
   switching the busbar configuration, we can reroute power flows
   through other lines that may have more capacity, thus reducing the
    risk of overloading any single line. Given the information, a
   potential action could be to change the busbar configuration at
   either substation 5 or substation 12 to redistribute the load
   across other lines more evenly. However, without specific details
   on the connectivity and capacity of adjacent lines or the impact
   of busbar changes on the overall grid topology, recommending a
   precise action is challenging.

For demonstration purposes, let's assume changing the busbar
   configuration at substation 5 for one of the connected elements
   could help redistribute the load. This action is hypothetical and
   would typically require a detailed analysis of the grid's topology
```

```
    and simulations to ensure it doesn't inadvertently isolate parts
   of the grid or overload other lines. This action and rationale are
    based on the provided grid state and the assumption that changing
    the busbar configuration can help manage line loading. In
   practice, a detailed analysis and possibly real-time simulations
   would be necessary to identify the best course of action.
'''
```

The agent identifies the correct connections between highly loaded lines and their corresponding substations and proposes changing the topologies at adjacent substations, which nominally sounds like a valid strategy. In terms of identifying unusual state values and predicting failures, the naive LLM agent performs well. However, even with sound logic and identifying promising topology changes, the LLM performs poorly in benchmarks. To see a full response over multiple timesteps for the agent, consult Appendix B.

Nested ReAct Agent with Tools

For the nested ReAct agent, we can see that it also successfully parses the topology observation and chooses the tools that would be most helpful. For this cascading failure scenario, it is able to identify the overloaded lines, call the `RankTopology` tool with the most overloaded line as the input, and decide that the action is suitable to submit.

```
---ANALYZING OBSERVATION---
"The current observation shows three overloaded lines with IDs 9, 16,
    and 14, ranked from most to least critical based on their
   loading. Line 9 is the most overloaded, followed by line 16 and
   line 14. All other aspects of the grid, including substations,
   generators, and loads, appear to be operating within normal
   parameters. Given the presence of overloads, it\'s advisable to
   take action to mitigate these overloads and prevent potential
   cascading failures. The \'do_nothing\' action is available but may
    not be the best choice given the current overloads."
---DECIDING TOOL OR ACTION---
---DECISION: TOOL---
"Given the current overloads on lines 9, 16, and 14, it\'s critical
   to address the most overloaded line first to prevent a cascading
   failure. Using the \'generate_ranked_topology_action\' tool will
   allow us to simulate different topologies for the most critical
   line (line 9) and choose an action that could potentially
   alleviate the overload by redistributing the flow or changing the
   grid\'s configuration in a beneficial way."
<Agent chose generate_ranked_topology_action>
"To address the most critical overload on line 9, generating a ranked
    topology action for this line is the most direct approach to
   potentially alleviate the overload. This action can help
   redistribute the load and prevent a cascading failure, which is
   crucial for maintaining grid stability."
---RANK TOPOLOGIES TOOL CALL---
```

```
"The topology ranking system computed an action that solves all
   overloads by changing topology at substation 3. For future
   reference, the action is called topology_suggestion_9."
---ANALYZING OBSERVATION---
"The topology ranking system has generated an action named \'
   topology_suggestion_9\' that solves all overloads by changing the
   topology at substation 3. This action appears to be a direct
   response to the critical overloads identified on lines 9, 16, and
   14. Given this solution directly addresses the current grid
   challenges, it is advisable to proceed with this action rather
   than the \'do_nothing\' option. Submitting \'topology_suggestion_9
   \' could alleviate the overloads and enhance grid stability."
---DECIDING TOOL OR ACTION---
---DECISION: ACTION---
"The action \'topology_suggestion_9\' directly addresses the critical
    overloads on lines 9, 16, and 14 by changing the topology at
   substation 3. This proactive measure is expected to redistribute
   the load effectively, alleviating the overloads and enhancing the
   overall stability of the grid. Therefore, it is the most suitable
   action to take at this moment to prevent potential cascading
   failures and ensure the grid operates within safe parameters."
<Agent chose topology_suggestion_9>
---SUBMITTING ACTION---
This action will:
         - NOT change anything to the injections
         - NOT perform any redispatching action
         - NOT modify any storage capacity
         - NOT perform any curtailment
         - NOT force any line status
         - NOT switch any line status
         - NOT switch anything in the topology
         - Set the bus of the following element(s):
                 - Assign bus 2 to line (extremity) id 3 [on
                    substation 3]
                 - Assign bus 2 to line (extremity) id 5 [on
                    substation 3]
                 - Assign bus 1 to line (origin) id 6 [on substation
                    3]
                 - Assign bus 2 to line (origin) id 15 [on substation
                     3]
                 - Assign bus 1 to line (origin) id 16 [on substation
                     3]
                 - Assign bus 1 to load id 2 [on substation 3]
Timestep summary: The action to change the topology at substation 3
   has been submitted. This action reconfigures the bus assignments
   for specific lines and a load within substation 3, aiming to
   alleviate the overloads on lines 9, 16, and 14 by optimizing the
```

```
flow of electricity through the grid. This strategic move is
expected to enhance grid stability by redistributing the load more
 evenly across the network, thus addressing the critical overloads
 identified in the previous step.
```

3.2 Scores

The scores for the agents across the different simulation environments are displayed in 3.1 and 3.2. It is important to recognize that, due to compute and cost limitations, these scores were only computed over 12 timesteps in each scenario, which represents one hour of operation data. This means that the full breadth of variations in generation, loads, and attacks on the grid were not simulated, and an area of future work would be focused on evaluating over longer episodes when more compute is available.

Table 3.1: L2RPN Benchmark Reward Across Agents For Steady State Operation

Timestep	**Agent**				
	DoNothing	**Random**	**Expert**	**ReAct**	**ReAct with Tools**
0	22.88	41.39	23.62	29.28	26.92
1	21.86	42.39	23.30	28.41	27.49
2	22.23	-	19.82	29.07	27.22
3	22.26	-	19.33	28.67	26.99
4	22.09	-	19.15	28.74	26.79
5	20.99	-	21.51	27.87	26.41
6	21.10	-	21.01	27.98	26.33
7	20.75	-	18.15	-	25.88
8	19.54	-	18.32	-	26.07
9	20.25	-	18.11	-	25.20
10	19.06	-	18.21	-	25.17
11	18.72	-	23.62	-	23.83
Total	251.73	-	244.15	-	314.29

Note: "-" indicates the episode terminated prematurely due to grid failure.

The scores for the steady-state operation scenario in Table 3.1 demonstrate how the LLM agents perform poorly when there is no immediate threat to grid stability. It is hypothesized that this may be due to OpenAI's post-training that caused the LLMs to return very conservative outputs, often doing nothing if there are no overloaded lines. The naive ReAct agent fails 35 minutes into the simulation because the topology changes it submitted pushed the grid away from steady-state operation, in a similar manner to the Random Agent. This demonstrates the need for LLMs to have access to simulation tools to understand the effects of their actions. The nested ReAct agent which had access to a simulator managed to complete the hour-long simulation, however it still had higher energy losses than the DoNothing and Expert agents. This could be due to multiple factors, like how the tools were

Table 3.2: L2RPN Benchmark Reward Across Agents For Cascading Failure Scenario

Timestep	**Agent**				
	DoNothing	**Random**	**Expert**	**ReAct**	**ReAct with Tools**
0	37.97	76.73	36.20	28.99	24.28
1	38.16	-	26.10	29.48	23.50
2	-	-	21.53	-	24.00
3	-	-	21.48	-	23.69
4	-	-	21.48	-	22.42
5	-	-	20.75	-	21.41
6	-	-	20.94	-	19.89
7	-	-	19.96	-	20.01
8	-	-	19.25	-	19.93
9	-	-	19.14	-	19.31
10	-	-	18.62	-	19.14
11	-	-	18.52	-	18.69
Total	-	-	263.97	-	256.29

designed to reduce overloads, or how the prompts emphasized the main goal of surviving the simulation while not putting enough emphasis on the secondary goal of minimizing economic losses for the operator.

The cascading failure scenario provides more promising results for the tool-use agent. Since the simulation starts with two critical lines being disconnected in the 14-bus grid, other lines immediately begin getting overloaded. The Random agent exacerbates the issue and fails within this first timestep, while the DoNothing and naive ReAct agents survive an additional timestep as the lines became increasingly overloaded. Both the Expert agent and the LLM agent with tools survive the entire hour, but the LLM agent performs slightly better due to taking a more drastic topology change action in the first timestep. The scores between the Expert and ReAct with Tools agent are too close to draw any solid conclusion, however.

Chapter 4

Conclusion

4.1 Discussion

The results paint a complex picture. LLMs are exceptional at understanding higher-level tasks and performing step-by-step reasoning to complete their goal, but their interaction with the environment needs to be carefully designed. In the naive LLM agent design, the LLM was given the ability to choose any action in the action space, but it struggled to find advantageous topologies because it lacked access to a simulator. For the LLM agent with tools, providing it with tools made it constrained by their outputs, stifling its creativity and causing it to fail when the tools failed to provide useful results. It can be concluded that there are two main reasons for the agents' results: the tool design and the level of domain expertise of LLMs. The tools were designed for averting critical failure scenarios, and many of them returned useful results only when presented with extreme grid observations. For example, the `RankTopology` tool required an overloaded line as input before it could simulate topology solutions, and would return an unhelpful response if the provided line had a loading percentage under 100%. The tool was based on a wrapper for ExpertOp4Grid, which was designed to remediate extreme line loading scenarios. During evaluation, the LLM agent attempted to rank topologies even during steady-state operation to no avail, and this can be seen as a limitation by the toolset rather than by the LLM, which points to why scores were better in the cascading failure scenario compared to the steady-state one. Additionally, though the LLM agents exhibited understanding of the operation task and could parse the topology, they could not perform further analyses due to the lack of power systems data in their training set. Apart from very general operations principles, the LLM could not explain its chosen actions with power flow estimates or historical operations data. A potential solution for this issue is improved prompting through Retrieval-Augmented Generation (RAG), where prompts are augmented with specific power flow results from a database, or by fine-tuning the LLM on datasets of power flow simulator results paired with their resulting operation decisions and outcomes. This difference in performance between steady-state and cascading failure scenarios could also point to how rule-based systems, like the Expert Agent, perform well during routine tasks, but in dynamic scenarios that require exploring multiple different paths, LLMs can provide value.

4.2 Limitations

This work was limited by available compute during the development and evaluation process, and therefore only closed-source models on the cloud were utilized. The cost of such a method necessitated shorter episode lengths during evaluation, which limits the amount of insight that can be gleaned from just a few hour-long scenarios. For the naive ReAct agent, a single timestep consisted of one call to the OpenAI server, with at minimum 1500 input tokens comprising the grid topology and role specification, and around 250 output tokens for the LLM's JSON response and rationale for the chosen action. At the time of writing, this single call costed around 10 cents. The shortest episode length in the original L2RPN benchmark is two days, or 576 timesteps, which would cost 57.60 USD. The nested ReAct agent has a significantly higher cost, due to the extra calls to the LLM required to choose tools with input parameters, parse tool outputs, and perform additional reasoning. For this thesis, there was a hard limit set to 20 messages, or about 7 calls to the LLM for each timestep. Due to previous messages being included in each new prompt to maintain context, the number of input tokens would increase linearly between calls in a single timestep, and therefore the cost per timestep would increase exponentially with the number of tool calls. For the 12-timestep scenarios in this thesis, a single run costed on the order of 20 USD.

Attempts were made to use open-source models that could be loaded locally on Google Colaboratory's Pro tier, however the open-source models that were feasible within the compute and budget constraints were unable to consistently perform the task. Colaboratory's Pro tier allows for 15 GB of GPU RAM, which is enough to load a 4-bit quantized version of Mistral-7B, however without extra fine-tuning on the power systems domain and more advanced memory-optimization techniques to preserve context across tool-calls and multiple timesteps, the models failed to return responses that conformed to the schemas required to proceed with the simulation. There is potential to remedy this issue by using more advanced prompting and parsing techniques, or a Mixture of Experts (MoE) architecture to increase the likelihood of valid responses.

4.3 Future Work

This thesis leaves many open questions and as such there are quite a few avenues to research. The use of actions that could redispatch generators, curtail renewable sources, and send timely alerts of failure, all features of the most recent L2RPN competitions, were out of scope for this thesis, but would be a valuable next step to evaluate. Topological actions are a large but discrete action space, while redispatch and curtailment are continuous action spaces which may be better suited for LLMs. Additionally, when an agent understands it will not be able to resolve overloads, sending timely alerts are essential to garnering the trust of human operators, and implications of how LLMs would perform on this task must be tested. Once the LLM agents are able to successfully recommend topological actions on the IEEE Case 14 grid, it would be important to benchmark them on the other Grid2Op environments, such as the WCCI 2020, NeurIPS 2020 Track 1, and NeurIPS 2020 Track 2 grids. Because all the open-sourced winning agents were trained on the data in these existing environments, it was not possible to evaluate them against the LLM agents and other baselines, so it would also be important to create new environments with realistic timeseries outside of these training sets to perform this evaluation. For future reproduceability, it would also be valuable to use an open-source LLM like Mistral-7B or LLaMA-7B, which can be deployed on local GPU clusters. Before deploying to real-life scenarios, there needs to be an evaluation of the effectiveness of LLM agents in interfacing with grid operators. Though providing status updates, like the current implementation, helps the human user understand the LLM's train of thought, it fails to give any guarantees about confidence or accuracy. Thorough testing needs to be done to prevent hallucinations by the LLM and to win the trust of control room operators [15]. Nevertheless, LLM agents have the potential to revolutionize how 21st century grids are operated and to improve the reliability, resilience, and our understanding of power systems.

References

[1] A. Marot, A. Kelly, M. Naglic, V. Barbesant, J. Cremer, A. Stefanov, and J. Viebahn. "Perspectives on Future Power System Control Centers for Energy Transition". In: *Journal of Modern Power Systems and Clean Energy* 10.2 (2022), pp. 328–344. DOI: 10.35833/MPCE.2021.000673.

[2] Z. Song, C. M. Hackl, A. Anand, A. Thommessen, J. Petzschmann, O. Kamel, R. Braunbehrens, A. Kaifel, C. Roos, and S. Hauptmann. "Digital Twins for the Future Power System: An Overview and a Future Perspective". In: *Sustainability* 15.6 (2023). ISSN: 2071-1050. DOI: 10.3390/su15065259. URL: https://www.mdpi.com/2071-1050/15/6/5259.

[3] N. A. E. R. Corporation. *2024 State of Reliability: Overview*. Washington, DC: North American Electric Reliability Corporation, June 2024.

[4] R. Gitzel et al. "Toward Cognitive Assistance and Prognosis Systems in Power Distribution Grids—Open Issues, Suitable Technologies, and Implementation Concepts". In: *IEEE Access* 12 (2024), pp. 107927–107943. DOI: 10.1109/ACCESS.2024.3437195.

[5] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[6] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. *ReAct: Synergizing Reasoning and Acting in Language Models*. 2023. arXiv: 2210.03629 [cs.CL]. URL: https://arxiv.org/abs/2210.03629.

[7] A. Marot, B. Donnot, G. Dulac-Arnold, A. Kelly, A. O'Sullivan, J. Viebahn, M. Awad, I. Guyon, P. Panciatici, and C. Romero. "Learning to run a Power Network Challenge: a Retrospective Analysis". In: *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*. Ed. by H. J. Escalante and K. Hofmann. Vol. 133. Proceedings of Machine Learning Research. PMLR, June 2021, pp. 112–132. URL: https://proceedings.mlr.press/v133/marot21a.html.

[8] T. Lan, J. Duan, B. Zhang, D. Shi, Z. Wang, R. Diao, and X. Zhang. "AI-Based Autonomous Line Flow Control via Topology Adjustment for Maximizing Time-Series ATCs". In: *2020 IEEE Power Energy Society General Meeting (PESGM)*. 2020, pp. 1–5. DOI: 10.1109/PESGM41954.2020.9281518.

[9] B. Zhou, H. Zeng, Y. Liu, K. Li, F. Wang, and H. Tian. *Action Set Based Policy Optimization for Safe Power Grid Management*. June 2021. DOI: 10.48550/arXiv.2106.15200.

[10] Z. Yan and Y. Xu. “Real-Time Optimal Power Flow With Linguistic Stipulations: Integrating GPT-Agent and Deep Reinforcement Learning”. In: *IEEE Transactions on Power Systems* 39.2 (2024), pp. 4747–4750. DOI: 10.1109/TPWRS.2023.3338961.

[11] H. F. Hamann et al. “Foundation models for the electric power grid”. In: *Joule* 8.12 (2024), pp. 3245–3258. ISSN: 2542-4351. DOI: https://doi.org/10.1016/j.joule.2024.11.002. URL: https://www.sciencedirect.com/science/article/pii/S2542435124004707.

[12] B. Donnot. *Grid2op- A testbed platform to model sequential decision making in power systems.* 2020. URL: %5Curl%7Bhttps://GitHub.com/Grid2Op/grid2op%7D.

[13] A. Marot, B. Donnot, C. Romero, L. Veyrin-Forrer, M. Lerousseau, B. Donon, and I. Guyon. *Learning to run a power network challenge for training topology controllers*. 2019. arXiv: 1912.04211 [eess.SP]. URL: https://arxiv.org/abs/1912.04211.

[14] A. Marot, B. Donnot, S. Tazi, and P. Panciatici. “Expert system for topological remedial action discovery in smart grids”. In: *Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2018)*. 2018, pp. 1–6. DOI: 10.1049/cp.2018.1875.

[15] J. Ruan, G. Liang, H. Zhao, G. Liu, X. Sun, J. Qiu, Z. Xu, F. Wen, and Z. Y. Dong. “Applying Large Language Models to Power Systems: Potential Security Threats”. In: *IEEE Transactions on Smart Grid* 15.3 (2024), pp. 3333–3336. DOI: 10.1109/TSG.2024.3373256.

Appendix A

Code listing

A.1 Prompts

A.1.1 Naive LLM System Prompt

```
"""You are a power systems expert challenged to test your skills with
    grid2op, a power grid simulator.
Your objective is to successfully manage to operate a powergrid,
   meaning you can change the topology to make sure it remains safe
   while being operated to minimize the energy losses.
The simulation is turn-based, so you will observe the environment,
   submit an action, and wait for the next observation.

You will immediately fail if the power grid isn't operating properly,
    defined by the following two conditions:
1. Consumption is not met because no electricity is flowing to some
   loads or more than one power plant gets disconnected.
2. The grid gets split apart into isolated sub-grids making the whole
    grid non-convex.

These conditions can appear when power lines in the grid get
   disconnected after being overloaded. When a line get disconnected,
    it loads gets distributed over other power lines, which in turn
   might get overloaded and thus disconnected as well, leading to a
   cascading failure (blackout).
When the power in a line increases above its thermal limit, the line
   becomes overloaded. It can stay overloaded for few timesteps
   before it gets disconnected, if no proper action is taken to
   relieve this overload within 2 timesteps (10 minutes of real time)
   , this is what we call a "soft overload".
If the overload is too high, the line gets disconnected immediately (
   above 200% of the thermal limit). This is a 'hard' overload. At
   some point this can lead to a very rapid cascading failure in a
```

```
   single timestep, if some lines already got disconnected and other
   lines get quite loaded.

At each timestep, you'll be given current productions, loads, and
   more importantly the flows over the lines and the topology of the
   grid. You will then have to choose to perform zero, one, or
   multiple actions.

Actions can consist of:
- Disconnecting or reconnecting a powerline
- Changing the topology of the grid (choose to isolate some objects [
   productions, loads, powerlines] from others by switching their
   busbar at their substation)
Be aware that some actions can be considered illegal by grid2op if
   they don't comply with some conditions. In that case, no action
   will be taken at that timestep, similar to a do-nothing action.

You will be evaluated on the cost of operations of a power grid, as
   well as the cost of any blackout that could occur.
At any time, the operator of the grid is responsible for compensating
    line losses by purchasing on the energy market the corresponding
   amount of production at the marginal price.

You will be evaluated over an entire scenario, and if you immediately
    fail due to blackouts or the fail conditions previously mentioned
   , the cost of blackout is calculated from the time of blackout to
   the end of the scenario, which will be significantly higher than
   any other operational cost.

Valid actions to return:
{{}} means no action.
{{"set_line_status": [{{"id": <line_id>, "status": <status>}}]}}
   where <line_id> is an integer and <status> is `+1` or `-1`
   indicating on or off.
{{"set_bus": [{{"id": <sub_id>, "target": <topo_target>}}]}} where <
   sub_id> and <topo_target> are integers.

Do not use placeholder strings like 'gen_id', 'line_id', or empty
   values like {{"set_bus": {{}}}}.

You are also responsible for justifying your choice of action with a
   rationale.

Return only JSON following this schema:
{{
  "action": "<action>",
  "rationale": "<rationale>"
```

```
}}
"""
```

A.1.2 Nested ReAct LLM with Tools System Prompt

```
"""
You are a power systems expert assistant challenged to test your
   skills with grid2op, a power grid simulator.
Your objective is to successfully manage to operate a powergrid,
   meaning you can change the topology to make sure it remains safe
   while being operated to minimize the energy losses.
The simulation is turn-based, so you will observe the environment,
   submit an action, and wait for the next observation.

You will immediately fail if the power grid isn't operating properly,
    defined by the following two conditions:
1. Consumption is not met because no electricity is flowing to some
   loads or more than one power plant gets disconnected.
2. The grid gets split apart into isolated sub-grids making the whole
    grid non-convex.

These conditions can appear when power lines in the grid get
   disconnected after being overloaded. When a line get disconnected,
    it loads gets distributed over other power lines, which in turn
   might get overloaded and thus disconnected as well, leading to a
   cascading failure (blackout).
When the power in a line increases above its thermal limit, the line
   becomes overloaded. It can stay overloaded for few timesteps
   before it gets disconnected, if no proper action is taken to
   relieve this overload within 2 timesteps (10 minutes of real time)
   , this is what we call a "soft overload".
If the overload is too high, the line gets disconnected immediately (
   above 200 percent of the thermal limit). This is a 'hard'
   overload. At some point this can lead to a very rapid cascading
   failure in a single timestep, if some lines already got
   disconnected and other lines get quite loaded.

At each timestep, you'll be given current productions, loads, and
   more importantly the flows over the lines and the topology of the
   grid. You will then have to choose to perform zero, one, or
   multiple actions.

Actions can consist of:
- Disconnecting or reconnecting a powerline
- Changing the topology of the grid (choose to isolate some objects [
   generators, loads, powerlines] from others by switching which of
```

```
   the two busbars at their substation they are connected to)
Be aware that some actions can be considered illegal by grid2op if
   they don't comply with some conditions. In that case, no action
   will be taken at that timestep, similar to a do-nothing action.

You will be evaluated on the cost of operations of a power grid, as
   well as the cost of any blackout that could occur.
At any time, the operator of the grid is responsible for compensating
    line losses by purchasing on the energy market the corresponding
   amount of production at the marginal price.

You will be evaluated over an entire scenario, and if you immediately
    fail due to blackouts or the fail conditions previously mentioned
   , the cost of blackout is calculated from the time of blackout to
   the end of the scenario, which will be significantly higher than
   any other operational cost.

You must return a valid Grid2Op action at each timestep. If you
   return an invalid action, the environment will return a "do
   nothing" action. You will start out with only the "do nothing"
   action available to you, and you will need to generate new actions
    using the tools at your disposal.
Do not try to synthesize the action yourself, instead use the tools
   at your disposal to generate actions and choose from those. Some
   tools will let you combine actions and simulate them, so take
   advantage of those.
Tools:
"rank_overloads" - Ranks overloaded lines by criticality and usage
   rate.
"generate_ranked_topology_action" - Simulates topologies for an
   overloaded line of your choosing and ranks them based on effect on
    line overloads, returning the best ranking action.
"generate_reference_topology_action" - Generates an action that
   restores a substation of your choosing to its reference topology.
"generate_reconnect_line_action" - Generates an action to reconnect a
    line of your choosing that was previously disconnected.
"generate_combined_action" - Generates an action that combines
   multiple previously generated actions of your choosing.
"simulate_action" - Simulates the effects of an action on the grid,
   validates that it is a legal action, and reports the line loadings
    after simulating.

When there are overloaded lines, a typical human operator usually
   chooses actions that solve the most critical overloads from high-
   ranked topologies. If there are no overloads or they are not
   satisfied with the high-ranked topology actions, they will choose
   actions to recover the busbar topologies of substations that were
```

```
    previously changed or reconnect disconnected lines to achieve a
    safer grid state and reduce line loading. However, you're free to
    try any combination of tools and actions you think will work best.
"""
```

Appendix B

Cascading Failure Scenario

B.1 Naive ReAct Agent

Timestep 0

```
Given the current state of the grid, there are no immediate signs of
   overload or disconnection that would necessitate urgent action.
   However, to optimize the grid's performance and minimize the risk
   of future issues, it's beneficial to address any potential
   vulnerabilities.

One area of concern is line 16, which has a loading of 1.1214734 (or
   112.14734% of its thermal limit), indicating it is currently
   overloaded. This situation, if not addressed, could lead to a "
   soft overload" and eventually a disconnection if the load
   continues to exceed the thermal limit for more than two timesteps.
    To prevent this, redistributing the load to alleviate stress on
   line 16 is advisable.

A potential action to mitigate this overload without detailed
   information on the capacity and current load of neighboring lines
   or the flexibility of generation units is to adjust the topology
   to redistribute the load more evenly across the network. However,
   without specific details on the network's current constraints and
   capacities, recommending a precise action (such as reconfiguring a
    substation or disconnecting a line) is challenging.

Given the simulator's constraints and the information provided, a
   general approach would be to explore reconfiguring the topology at
    substations connected to line 16 to redistribute the load. This
   could involve switching the busbar configurations at substations 3
    and 8 to alter the flow paths within the grid, potentially
   reducing the load on line 16.
```

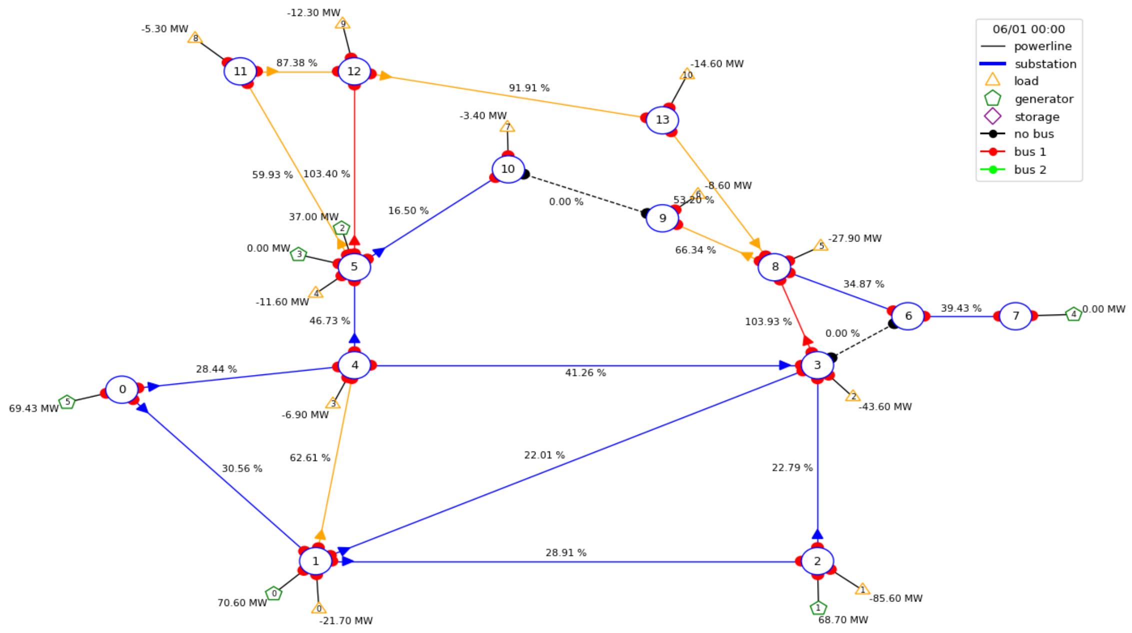


Figure B.1: Initial Line Failures

```
However, without the ability to simulate or predict the exact outcome
    of such actions within this response format, and to ensure
   compliance with the simulator's rules, a specific action cannot be
    confidently recommended without further analysis.

Therefore, the most prudent course of action in this scenario, given
   the limitations, is to monitor the situation closely and prepare
   to adjust the topology or generation dispatch as soon as more
   detailed information becomes available or if the simulator
   provides feedback indicating a specific overload or system
   constraint that can be directly addressed.

Given the current state of the grid, there is an overload on line 16,
    suggesting a need for action to redistribute the load. However,
   without detailed information on the network's constraints,
   capacities, and the ability to predict the outcome of specific
   actions, a precise action cannot be confidently recommended. The
   suggested approach is to monitor the grid closely and be prepared
   to adjust the topology or generation dispatch based on more
   detailed feedback or analysis.
This action will:
         - NOT change anything to the injections
         - NOT perform any redispatching action
         - NOT modify any storage capacity
         - NOT perform any curtailment
         - NOT force any line status
         - NOT switch any line status
```

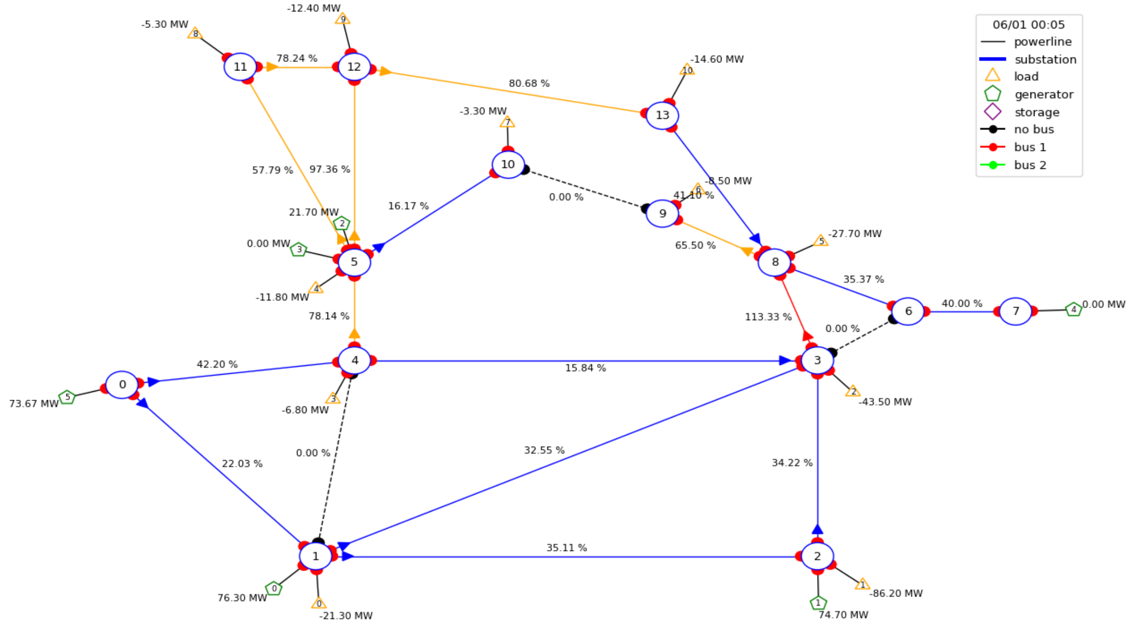


Figure B.2: Grid state after action at timestep 0

```
        - NOT switch anything in the topology
        - NOT force any particular bus configuration
```

Timestep 1

```
Based on the current state of the grid, we observe that most of the
   power lines are operating within safe limits, except for line 16
   (‘’3_8_16’‘), which has a loading of 113.32763%, significantly
   above the thermal limit. This indicates a hard overload condition,
    risking immediate disconnection and potentially triggering a
   cascading failure if not addressed promptly.

To mitigate this risk and prevent a potential blackout, the following
    action is proposed:

- **Action**: Reduce the load on line 16 by redistributing the flow.
   This can be achieved by changing the topology at substations
   connected to the overloaded line. Specifically, we can switch the
   busbar of some components (generators or loads) at substations 3
   and 8 to redistribute the power flow and relieve the overloaded
   line.

Given the complexity of the grid and the need for a precise solution
   that doesn’t inadvertently isolate parts of the grid or cause
   further overloads, a detailed analysis of the grid’s topology and
   current flows is required to determine the exact components to
```

````
switch. However, a general approach would involve examining the connected generators and loads at substations 3 and 8 and considering switching the bus of one or more components to balance the load more evenly across the network.

Without specific details on the grid's capacity for redistribution or the flexibility of connected components, a generic action cannot be accurately formulated. Therefore, the proposed action in JSON format is conceptual and aims to illustrate the type of intervention that might be considered:

```json
{
  "action": "To be determined based on a detailed analysis of the grid's current state and the capacity for load redistribution at substations 3 and 8.",
  "rationale": "The immediate action is required to address the hard overload on line 16 ('3_8_16') by redistributing the power flow to prevent its disconnection and avoid a cascading failure. The specific action, such as changing the topology by switching busbars at substations 3 and 8, needs to be carefully analyzed to ensure it effectively reduces the load on the affected line without causing unintended consequences on the grid's stability."
}
```

This response highlights the need for a dynamic and informed approach to grid management, where decisions are based on real-time data and a thorough understanding of the grid's operational capabilities and constraints.
The immediate action is required to address the hard overload on line 16 ('3_8_16') by redistributing the power flow to prevent its disconnection and avoid a cascading failure. The specific action, such as changing the topology by switching busbars at substations 3 and 8, needs to be carefully analyzed to ensure it effectively reduces the load on the affected line without causing unintended consequences on the grid's stability.
Failed to extract action:  unterminated string literal (detected at line 1) (<unknown>, line 1)
This action will:
         - NOT change anything to the injections
         - NOT perform any redispatching action
         - NOT modify any storage capacity
         - NOT perform any curtailment
         - NOT force any line status
         - NOT switch any line status
````

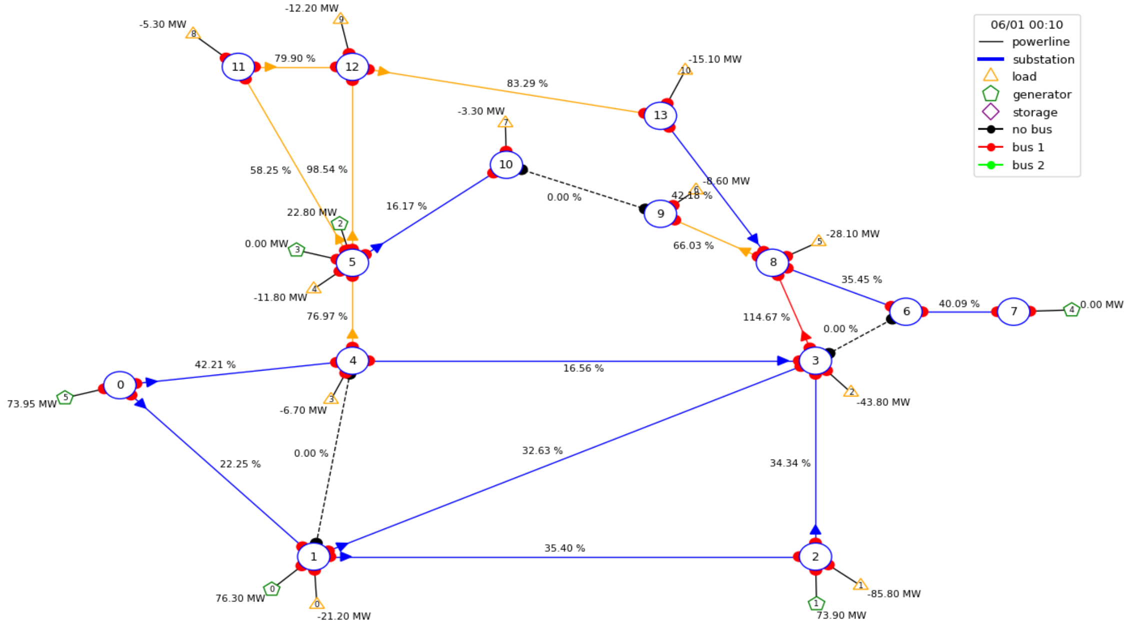


Figure B.3: Grid state after action at timestep 1

```
        - NOT switch anything in the topology
        - NOT force any particular bus configuration
```

Timestep 2

```
Given the current state of the grid, it's crucial to analyze the key
   components that could lead to potential failures or
   inefficiencies. The primary focus should be on managing line
   loadings and ensuring that no line is at risk of a hard or soft
   overload, which could lead to disconnections and potentially a
   cascading failure.

Upon reviewing the provided grid state, the most concerning lines
   based on their loading percentages are:
- Line 9 (from substation 5 to substation 12) with a loading of 98
   .535454%, dangerously close to a hard overload condition.
- Line 16 (from substation 3 to substation 8) with a loading of 114
   .67079%, which is already in a hard overload condition and poses
   an immediate risk of disconnection.

To mitigate these risks and ensure the grid remains operational, the
   following actions are proposed:

1. **For Line 9 (5_12_9)**: This line is very close to a hard
   overload condition. To alleviate the load on this line,
   redistributing the power flow by adjusting the topology at
```

````
substations 5 or 12 could help. However, without specific details on the alternative paths for power flow or the capacity of adjacent lines, a direct action for this line is challenging to specify without risking unintended consequences.

2. **For Line 16 (3_8_16)**: This line is already in a hard overload condition and requires immediate action to prevent disconnection. Given the critical loading, disconnecting this line preemptively and rerouting the power through alternative paths might be necessary to prevent a cascading failure. However, this action should be taken with caution, ensuring that the rerouted power does not lead to overloads on other lines.

Given the complexity of the grid and the potential for cascading failures, a more detailed analysis of the grid's topology and the capacity of alternative routes is necessary to formulate a precise action plan. For the purpose of this simulation and based on the provided information, the proposed action is to attempt to redistribute the load by changing the topology at substations connected to the overloaded lines, specifically focusing on substation 5 (connected to Line 9) and substation 3 (connected to Line 16). However, without specific details on the grid's topology and alternative routes, a precise action cannot be confidently recommended.

Therefore, the recommended action is to monitor these lines closely in the next timestep and prepare to adjust the topology or disconnect lines as necessary based on the updated grid state. This approach is conservative, aiming to prevent immediate failure while acknowledging the need for more information to make a targeted intervention.

```json
{
  "action": "{}",
  "rationale": "Given the current grid state, with Line 16 in hard overload and Line 9 close to hard overload, the immediate action is to monitor these lines closely. A specific action, such as changing the topology or disconnecting a line, requires a more detailed analysis of the grid's topology and alternative routes to ensure that the intervention does not lead to unintended consequences. The decision to not take immediate action is based on the need for further analysis and to avoid precipitating a cascading failure by hastily rerouting power or disconnecting lines without a clear understanding of the grid's overall capacity and topology."
}
````
````

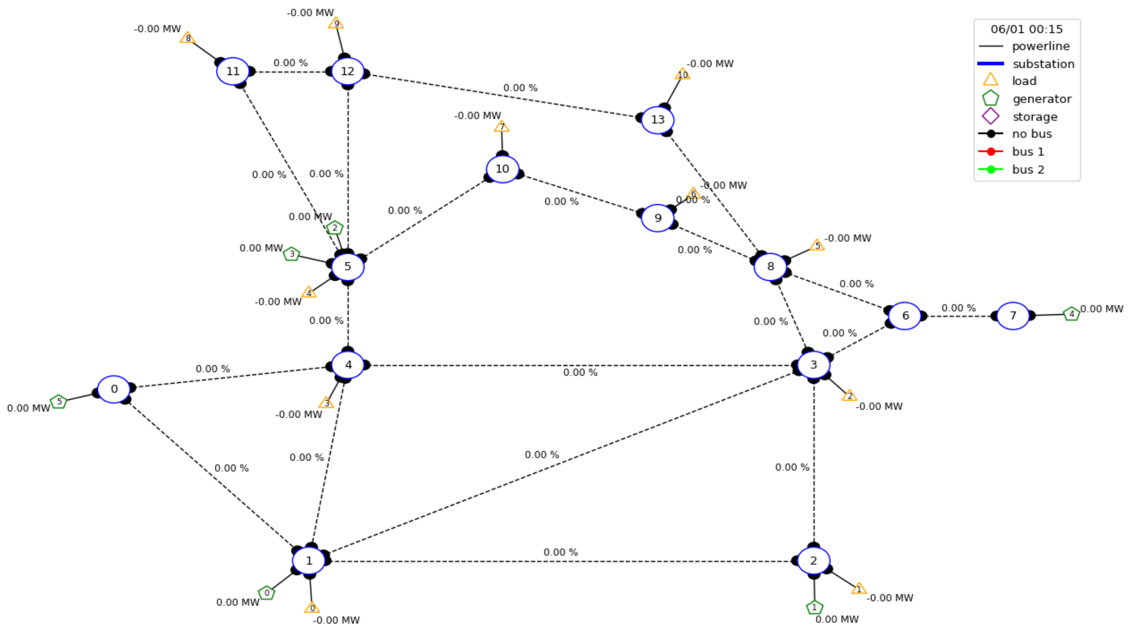


Figure B.4: Grid state after action at timestep 2

~~~
‘‘‘
Given the current grid state, with Line 16 in hard overload and Line
   9 close to hard overload, the immediate action is to monitor these
    lines closely. A specific action, such as changing the topology
   or disconnecting a line, requires a more detailed analysis of the
   grid’s topology and alternative routes to ensure that the
   intervention does not lead to unintended consequences. The
   decision to not take immediate action is based on the need for
   further analysis and to avoid precipitating a cascading failure by
    hastily rerouting power or disconnecting lines without a clear
   understanding of the grid’s overall capacity and topology.
This action will:
         - NOT change anything to the injections
         - NOT perform any redispatching action
         - NOT modify any storage capacity
         - NOT perform any curtailment
         - NOT force any line status
         - NOT switch any line status
         - NOT switch anything in the topology
         - NOT force any particular bus configuration
~~~

B.2 Nested ReAct Agent with Tools

Timestep 0

```
---ANALYZING OBSERVATION---
"The current observation shows three overloaded lines with IDs 9, 16,
    and 14, ranked from most to least critical based on their
   loading. Line 9 is the most overloaded, followed by line 16 and
   line 14. All other aspects of the grid, including substations,
   generators, and loads, appear to be operating within normal
   parameters. Given the presence of overloads, it\'s advisable to
   take action to mitigate these overloads and prevent potential
   cascading failures. The \'do_nothing\' action is available but may
    not be the best choice given the current overloads."
---DECIDING TOOL OR ACTION---
---DECISION: TOOL---
"Given the current overloads on lines 9, 16, and 14, it\'s critical
   to address the most overloaded line first to prevent a cascading
   failure. Using the \'generate_ranked_topology_action\' tool will
   allow us to simulate different topologies for the most critical
   line (line 9) and choose an action that could potentially
   alleviate the overload by redistributing the flow or changing the
   grid\'s configuration in a beneficial way."
<Agent chose generate_ranked_topology_action>
"To address the most critical overload on line 9, generating a ranked
    topology action for this line is the most direct approach to
   potentially alleviate the overload. This action can help
   redistribute the load and prevent a cascading failure, which is
   crucial for maintaining grid stability."
---RANK TOPOLOGIES TOOL CALL---
"The topology ranking system computed an action that solves all
   overloads by changing topology at substation 3. For future
   reference, the action is called topology_suggestion_9."
---ANALYZING OBSERVATION---
"The topology ranking system has generated an action named \'
   topology_suggestion_9\' that solves all overloads by changing the
   topology at substation 3. This action appears to be a direct
   response to the critical overloads identified on lines 9, 16, and
   14. Given this solution directly addresses the current grid
   challenges, it is advisable to proceed with this action rather
   than the \'do_nothing\' option. Submitting \'topology_suggestion_9
   \' could alleviate the overloads and enhance grid stability."
---DECIDING TOOL OR ACTION---
---DECISION: ACTION---
"The action \'topology_suggestion_9\' directly addresses the critical
    overloads on lines 9, 16, and 14 by changing the topology at
   substation 3. This proactive measure is expected to redistribute
   the load effectively, alleviating the overloads and enhancing the
   overall stability of the grid. Therefore, it is the most suitable
   action to take at this moment to prevent potential cascading
   failures and ensure the grid operates within safe parameters."
```

```
<Agent chose topology_suggestion_9>
---SUBMITTING ACTION---
This action will:
         - NOT change anything to the injections
         - NOT perform any redispatching action
         - NOT modify any storage capacity
         - NOT perform any curtailment
         - NOT force any line status
         - NOT switch any line status
         - NOT switch anything in the topology
         - Set the bus of the following element(s):
                 - Assign bus 2 to line (extremity) id 3 [on
                   substation 3]
                 - Assign bus 2 to line (extremity) id 5 [on
                   substation 3]
                 - Assign bus 1 to line (origin) id 6 [on substation
                   3]
                 - Assign bus 2 to line (origin) id 15 [on substation
                    3]
                 - Assign bus 1 to line (origin) id 16 [on substation
                    3]
                 - Assign bus 1 to load id 2 [on substation 3]
Timestep summary: The action to change the topology at substation 3
   has been submitted. This action reconfigures the bus assignments
   for specific lines and a load within substation 3, aiming to
   alleviate the overloads on lines 9, 16, and 14 by optimizing the
   flow of electricity through the grid. This strategic move is
   expected to enhance grid stability by redistributing the load more
    evenly across the network, thus addressing the critical overloads
    identified in the previous step.
```

Timestep 1

```
---ANALYZING OBSERVATION---
"No overloaded lines are currently detected in the grid, indicating a
    stable operation at this timestep. The action to change the
   topology at substation 3 in the previous step may have contributed
    to alleviating potential overloads, enhancing grid stability.
   However, it\'s important to remain proactive in managing the grid
   to prevent future overloads and ensure efficient operation. Given
   the current stable state, it might be beneficial to explore
   actions that could further optimize the grid\'s performance or
   prepare for potential future fluctuations in load or generation."
---DECIDING TOOL OR ACTION---
---DECISION: TOOL---
"To ensure the grid remains stable and to identify any potential
   overloads that might not be immediately apparent, it\'s prudent to
```

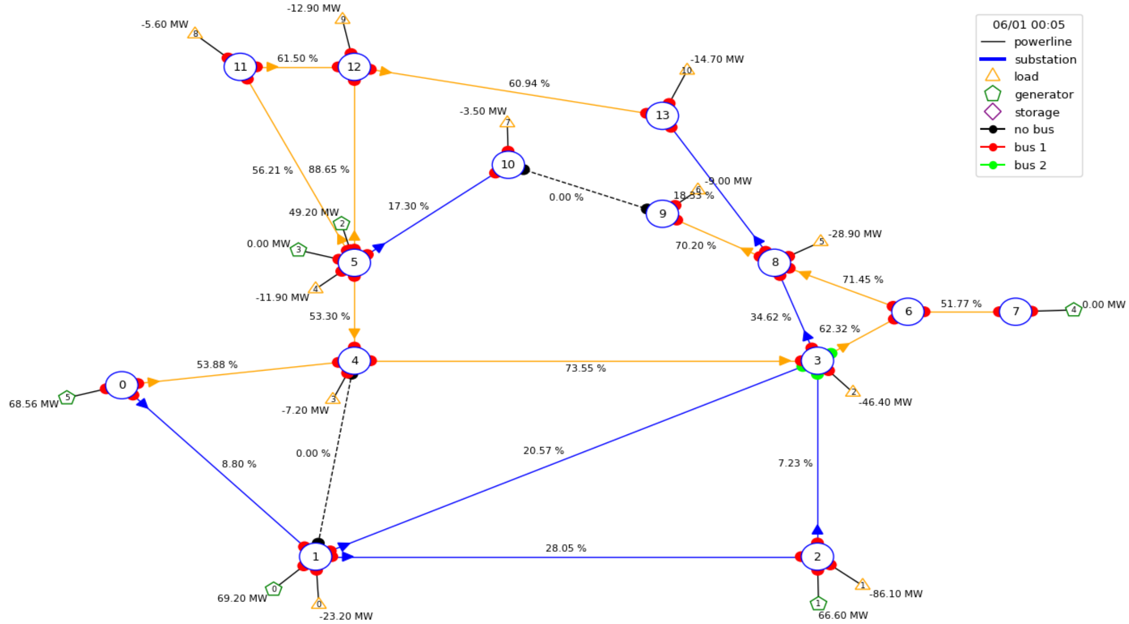


Figure B.5: Grid state after action at timestep 0

```
    rank overloads by criticality and usage rate. This will help in
   preemptively addressing any issues that could lead to overloads,
   ensuring the grid\'s efficient and reliable operation."
<Agent chose rank_overloads>
---RANK OVERLOADS TOOL CALL---
---ANALYZING OBSERVATION---
"Given the current grid state with no overloads and the previous
   action taken to adjust the topology at substation 3, the grid
   appears stable. The highest loading is observed on line 9 with a
   loading of 88.645%, which, while high, is not currently in an
   overload state. It\'s important to monitor such lines closely, as
   they are nearing their capacity and could become overloaded with
   further changes in the grid\'s state. Considering the grid\'s
   stability and the absence of immediate overloads, it might be
   strategic to either simulate potential actions that could further
   optimize the grid or prepare for future demand changes, or to
   proceed with a \'do_nothing\' action to maintain the current state
    while continuously monitoring the grid."
---DECIDING TOOL OR ACTION---
---DECISION: TOOL---
"Given the current stable state of the grid with no immediate
   overloads, it would be prudent to simulate the \'do_nothing\'
   action to ensure that maintaining the current configuration will
   not lead to any unforeseen issues in the immediate next timestep.
   This simulation will help confirm the grid\'s stability before
   making a decision to either take a specific action or continue
   with no changes."
```

```
<Agent chose simulate_action>
"Given the current grid state with no overloads and the stability
   achieved through previous actions, it is appropriate to proceed
   with a \'do_nothing\' action at this timestep. This decision is
   based on the absence of critical overloads and the need to observe
    the grid\'s performance following the recent topology change at
   substation 3. This approach allows for a cautious strategy,
   ensuring that the grid remains stable and efficient without
   unnecessary interventions. Monitoring the grid closely in
   subsequent timesteps will be crucial to respond promptly to any
   emerging issues."
---SIMULATE ACTION TOOL CALL---
---ANALYZING OBSERVATION---
"The simulation of the \'do_nothing\' action confirms the grid\'s
   stability, with no overloads detected. The line loads remain
   within acceptable limits, indicating that the current grid
   configuration is effectively managing the distribution of
   electricity. Given this stable state, it might be beneficial to
   continue monitoring the grid without making further changes at
   this timestep. This approach allows for the observation of any
   potential shifts in the grid\'s performance that may require
   intervention in future timesteps."
---DECIDING TOOL OR ACTION---
---DECISION: ACTION---
"The simulation results confirm that the grid remains stable with no
   overloads after simulating the \'do_nothing\' action. Given the
   current grid conditions and the absence of critical issues, it is
   appropriate to proceed with a \'do_nothing\' action for this
   timestep. This decision allows for the maintenance of the current
   stable state while conserving resources for any necessary
   interventions in the future. Continuous monitoring will be
   essential to promptly address any changes in the grid\'s
   performance."
<Agent chose do_nothing>
---SUBMITTING ACTION---
This action will:
         - NOT change anything to the injections
         - NOT perform any redispatching action
         - NOT modify any storage capacity
         - NOT perform any curtailment
         - NOT force any line status
         - NOT switch any line status
         - NOT switch anything in the topology
         - NOT force any particular bus configuration
Timestep summary: The "do_nothing" action has been successfully
   submitted to the environment, indicating a decision to maintain
   the current configuration of the power grid without making any
```

```
changes. This decision was based on the grid's current stability,
with no overloads detected and all line loads within acceptable
limits. The choice to not intervene allows the grid to continue
operating under its current stable state, while still being
prepared to take action in future timesteps if necessary.
Continuous monitoring of the grid's performance is essential to
ensure its ongoing stability and to address any potential issues
proactively.
```