LLM-Guided Multi-Agent Deep Reinforcement Learning for Distributed Energy Management in Renewable-Integrated Power Systems

Ruiqian Shi[1], Xiao Zhang[2], Yang Chen[1,*] and Hongwei Ding[3]

[1] Power China Chengdu Engineering Corporation Limited, Chengdu, China

[2] Jiujiang Polytechnic University of Science and Technology, Jiujiang, China

[3] School of Information Science and Engineering, Yunnan University, Kunming, China

INFORMATION

Keywords:
Large language model
multi-agent deep reinforcement learning
distributed energy management
renewable energy
smart grid
optimal scheduling

DOI: 10.23967/j.rimni.2026.10.76155

SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI

In cooperation with CIMNE

LLM-Guided Multi-Agent Deep Reinforcement Learning for Distributed Energy Management in Renewable-Integrated Power Systems

Ruiqian Shi[1], **Xiao Zhang**[2], **Yang Chen**[1,*] **and Hongwei Ding**[3]

[1]Power China Chengdu Engineering Corporation Limited, Chengdu, China

[2]Jiujiang Polytechnic University of Science and Technology, Jiujiang, China

[3]School of Information Science and Engineering, Yunnan University, Kunming, China

ABSTRACT

With the increasing penetration of renewable energy in power systems, distributed energy management faces numerous challenges including high-dimensional state spaces, multi-objective optimization, and real-time decision-making. This paper proposes a Large Language Model (LLM)-guided Multi-Agent Deep Reinforcement Learning (MADRL) framework for distributed energy management in renewable-integrated power systems. Building upon recent advances in LLM-guided reinforcement learning, we develop specialized mechanisms for power system control that leverage the semantic understanding and knowledge reasoning capabilities of LLMs to provide high-level strategic guidance and scenario-adaptive adjustments for MADRL agents. Specifically, we design a hierarchical architecture where the LLM layer is responsible for parsing grid operation states, generating optimization objective descriptions, and coordinating multi-agent behaviors, while the MADRL layer executes specific energy scheduling decisions. Experiments are conducted on real power grid datasets containing photovoltaic, wind power, energy storage systems, and flexible loads. Results demonstrate that the proposed method significantly outperforms traditional baseline methods in reducing operating costs, improving renewable energy utilization rates, and ensuring grid stability. Compared to standard MADRL, our method reduces system operating costs by 18.7%, decreases renewable energy curtailment by 23.4%, and improves convergence speed by 3.2 times. This study provides a novel approach for adaptive distributed energy management in smart grids.

OPEN ACCESS

Received: 15/11/2025

Accepted: 15/01/2026

Published: 16/04/2026

DOI
10.23967/j.rimni.2026.10.76155

Keywords:
Large language model
multi-agent deep reinforcement learning
distributed energy management
renewable energy
smart grid
optimal scheduling

1 Introduction

The global energy landscape is undergoing a profound transformation driven by the urgent imperative to decarbonize electricity systems and mitigate climate change. According to the International Energy Agency (IEA), renewable energy sources are projected to account for over 50% of

*Correspondence: Yang Chen (zswang_sub@163.com). This is an article distributed under the terms of the Creative Commons BY-NC-SA license

global electricity generation by 2030, with solar photovoltaic and wind power leading this unprecedented expansion. However, this rapid proliferation of variable renewable energy resources introduces formidable operational challenges to power grids that were originally designed for centralized, dispatchable generation. The inherent intermittency and unpredictability of solar and wind power—characterized by generation fluctuations spanning multiple temporal scales from seconds to seasons—create substantial complications for grid operators tasked with maintaining the critical balance between supply and demand while ensuring voltage stability, frequency regulation, and economic efficiency [1–3].

Traditional centralized energy management paradigms, which rely on predetermined dispatch schedules and reactive control mechanisms, are increasingly inadequate for addressing the complexities of modern renewable-integrated power systems. These conventional approaches struggle to accommodate the distributed nature of renewable generation, the bidirectional power flows introduced by prosumers (consumers who also produce energy), and the dynamic optimization requirements that emerge from real-time electricity price volatility [4–6]. Distributed energy management systems (DEMS) have emerged as a promising alternative, coordinating multiple heterogeneous distributed energy resources (DERs)—including photovoltaic arrays, wind turbines, battery energy storage systems, and flexible loads—to achieve system-wide optimization objectives through localized decision-making and peer-to-peer coordination [7,8]. However, the design of effective DEMS confronts several fundamental challenges: high-dimensional state and action spaces that grow exponentially with the number of DERs, multi-objective optimization involving conflicting goals such as cost minimization vs. renewable utilization maximization, real-time decision-making requirements under uncertainty, and the need for coordinated behaviors among multiple autonomous agents operating with partial observability [9,10].

Recent advances in artificial intelligence, particularly deep reinforcement learning (DRL), have demonstrated remarkable potential for addressing sequential decision-making problems in complex, uncertain environments [11–13]. By learning optimal policies through trial-and-error interactions with the environment, DRL methods can discover sophisticated control strategies without requiring explicit mathematical models of system dynamics—a critical advantage given the inherent unpredictability of renewable generation and load patterns [14–16]. Multi-agent deep reinforcement learning (MADRL) extends these capabilities to distributed settings where multiple learning agents must coordinate their actions to achieve collective objectives [17]. Several pioneering studies have successfully applied MADRL techniques to energy management scenarios, demonstrating substantial improvements over rule-based heuristics and model-based optimization approaches [18,19]. However, existing MADRL methods face persistent limitations that hinder their practical deployment in real-world power systems. First, these methods exhibit poor sample efficiency, requiring millions of training episodes to converge to near-optimal policies—a prohibitive requirement when training on actual hardware or high-fidelity simulators. Second, learned policies often lack generalization capability, performing well on training scenarios but degrading significantly when confronted with out-of-distribution conditions such as extreme weather events or equipment failures. Third, the coordination mechanisms in current MADRL approaches rely primarily on low-dimensional numerical message passing, which cannot convey the rich semantic information and causal reasoning necessary for sophisticated strategic coordination. Fourth, the "black-box" nature of deep neural network policies presents interpretability challenges, making it difficult for human operators to understand, trust, and intervene in automated control decisions [20].

Concurrently, large language models (LLMs) have achieved breakthrough capabilities in natural language understanding, commonsense reasoning, and knowledge integration across diverse domains

[21–23]. Pre-trained on vast corpora encompassing scientific literature, technical documentation, and general knowledge, LLMs possess remarkable zero-shot generalization abilities—they can tackle novel tasks described in natural language without task-specific fine-tuning [12]. Recent research has begun exploring the integration of LLMs with reinforcement learning for decision-making applications. For instance, Li et al. [17] proposed an LLM-guided decision-making toolkit for MARL that leverages language models to provide high-level strategic guidance, demonstrating improved sample efficiency in simulated environments. Yao et al. [13] developed a multi-agent fuzzy reinforcement learning framework enhanced with LLM for cooperative navigation of endovascular robotics, showing that semantic understanding facilitates inter-agent coordination. Zeng et al. [14] applied LLM-guided multi-agent approaches to chemical process optimization, achieving faster convergence through knowledge-based exploration. Samarathunga et al. [15] demonstrated that LLMs can enable natural language-based robot navigation by translating human instructions into executable policies. Sun and Li [24] utilized LLM guidance with representative agents for traffic modeling, illustrating the potential of language models in complex spatial-temporal optimization. Zhu et al. [16] investigated task offloading with LLM-enhanced MARL in UAV-assisted edge computing, reporting improved resource allocation efficiency. Yang et al. [25] explored LLM-guided reinforcement learning for interactive environments, highlighting the benefits of semantic reasoning in dynamic scenarios.

Despite these promising developments, the integration of large language models with multi-agent reinforcement learning for distributed energy management in renewable-integrated power systems remains an unexplored frontier. Existing applications of LLM-guided MARL focus predominantly on robotics navigation, process control, and network optimization domains, with little attention to the unique characteristics and requirements of power grid operations. Energy management presents distinct challenges that differ fundamentally from these application areas: strict safety constraints on voltage and frequency that must be satisfied at all times, multi-objective optimization involving economic, environmental, and stability considerations with dynamically varying priorities, operation across multiple temporal scales from real-time control (seconds) to scheduling (hours) and planning (days), and high-stakes decision-making where control errors can lead to equipment damage, service disruptions, or cascading blackouts. Furthermore, power systems possess rich domain knowledge—physical laws governing electrical networks, established operational principles, and best practices developed over decades—that can potentially be leveraged through LLMs' knowledge integration capabilities but has not been systematically exploited in prior MARL research.

Motivated by these observations and building upon the foundational work in LLM-guided reinforcement learning [13–17,24,25], this paper addresses the following fundamental research question: *Can large language models provide effective high-level semantic guidance to multi-agent reinforcement learning systems, enabling superior performance in distributed energy management for renewable-integrated power systems compared to existing approaches?* To answer this question, we propose a novel LLM-guided Multi-Agent Deep Reinforcement Learning (LLM-MADRL) framework that synergistically combines the complementary strengths of language models and reinforcement learning. Our key insight is that LLMs and MADRL address different aspects of the decision-making challenge: LLMs excel at semantic understanding, knowledge-based reasoning, and zero-shot generalization to novel scenarios, while MADRL specializes in learning optimal reactive policies through environmental interaction and multi-agent coordination. By establishing a hierarchical architecture where LLMs operate at a strategic level (scenario analysis, objective generation, coordination facilitation) and MADRL agents execute at a tactical level (real-time control actions), we hypothesize that the integrated system can achieve both superior performance and improved interpretability.

The specific contributions of this work are:

1. **Novel Framework Architecture:** We propose a comprehensive application of LLM-guided multi-agent reinforcement learning to distributed energy management in renewable-integrated power systems—a domain previously unexplored in the LLM-MARL literature. While the core techniques (LLM guidance, MADRL, hierarchical control) build upon prior work [13,14,17], our contribution lies in adapting and specializing these methods to address the unique challenges of power system operations: strict safety constraints, multi-timescale coordination, and domain-specific knowledge integration. Our hierarchical architecture decomposes the control problem into semantic reasoning (LLM cognition layer) and reactive decision-making (MADRL execution layer), enabling effective collaboration between symbolic and subsymbolic AI paradigms within the power systems context.
2. **LLM Guidance Mechanisms:** We design three specialized LLM-based guidance mechanisms that address key limitations of standard MADRL in energy management applications: (i) *scenario-aware objective generation* that dynamically adjusts optimization priorities based on contextual understanding of current grid conditions, (ii) *natural language communication protocol* that enables semantic coordination among agents beyond low-dimensional message vectors, and (iii) *dynamic reward shaping* that provides auxiliary learning signals informed by domain knowledge to accelerate policy learning and improve final performance. These mechanisms represent domain-specific instantiations of general LLM-RL integration principles, tailored to power system requirements.
3. **Comprehensive Experimental Validation:** We conduct extensive experiments on a realistic simulation environment constructed using IEEE 33-node distribution network topology with year-long operational data covering diverse seasons, weather conditions, and grid stress scenarios. Our evaluation encompasses six baseline methods, eight performance metrics across economic, renewable utilization, stability, and learning efficiency dimensions, and rigorous analysis of scenario adaptability, component contributions through ablation studies, computational efficiency, scalability to large systems, and generalization to out-of-distribution conditions.
4. **Significant Performance Improvements:** Experimental results demonstrate that LLM-MADRL achieves 33.3% cost reduction compared to rule-based baselines and 18.7% improvement over state-of-the-art MAPPO, while simultaneously increasing renewable energy utilization to 94.1% (vs. 90.7% for MAPPO), reducing voltage violations to 0.7% (vs. 1.2%), and accelerating convergence by 3.2× (620 episodes vs. 1980). Moreover, the approach exhibits superior robustness with only 8.3% performance degradation on out-of-distribution scenarios compared to 19.7% for MAPPO, and maintains advantages even when scaled to 30-agent systems.

The remainder of this paper is organized as follows: Section 2 surveys related work on distributed energy management, reinforcement learning in power systems, and the emerging integration of large language models with decision-making algorithms. Section 3 presents the problem formalization, system architecture, and detailed design of the LLM-MADRL framework including guidance mechanisms and training procedures. Section 4 reports comprehensive experimental results demonstrating the effectiveness of our approach across multiple evaluation dimensions. Section 5 concludes the paper.

2 Related Work

Our work sits at the intersection of three research areas: distributed energy management in renewable-integrated power systems, multi-agent reinforcement learning for control and optimization, and the emerging integration of large language models with decision-making algorithms. This section reviews relevant literature in each area and positions our contributions relative to existing work.

2.1 Distributed Energy Management in Renewable-Integrated Power Systems

The integration of renewable energy resources into power grids has necessitated the development of sophisticated energy management strategies capable of handling the variability, uncertainty, and distributed nature of solar and wind generation. Traditional centralized control approaches, which assume perfect observability and rely on mathematical optimization formulations, face fundamental challenges in modern grids characterized by high renewable penetration, prosumer participation, and dynamic operating conditions.

Early research on distributed energy management focused primarily on rule-based control strategies and model-based optimization techniques. Nougain and Panigrahi [1] proposed an integrated power management strategy for grid-tied DC microgrids that coordinates multiple distributed energy resources through hierarchical control architecture. While achieving reasonable performance under nominal conditions, rule-based approaches lack the adaptability required for highly variable renewable generation patterns and cannot optimize complex multi-objective trade-offs. Jin et al. [2] developed robust power management capabilities for integrated energy systems including linear and non-linear loads, demonstrating that advanced optimization formulations can improve performance but at the cost of increased computational complexity and sensitivity to model inaccuracies. Sun et al. [3] investigated distributed real-time power balancing in renewable-integrated grids with storage and flexible loads, highlighting the communication and coordination challenges that arise in decentralized control schemes. These model-based approaches require accurate system models and struggle with the inherent unpredictability of renewable sources.

Recent advances have focused on multi-objective optimization frameworks that explicitly balance competing goals such as economic efficiency, renewable utilization, and grid stability. Shafiei et al. [4] proposed a multi-objective optimization approach for resilience enhancement considering integrated energy systems with renewable energy, energy storage, energy sharing, and demand-side management. Their work demonstrated the importance of coordinated control across multiple DER types but relied on offline optimization that cannot adapt to real-time conditions. Tummuru et al. [5] developed dynamic energy management for renewable grid-integrated hybrid energy storage systems, showing that coordinated battery and supercapacitor control can smooth renewable variability. Saxena et al. [6] explored intelligent load forecasting and renewable energy integration for enhanced grid reliability, emphasizing the value of predictive capabilities in proactive energy management. However, these approaches remain fundamentally reactive and do not learn from experience to improve performance over time.

The distributed nature of modern power systems has motivated research on decentralized and distributed control architectures. Liu et al. [7] developed distributed robust energy management for multimicrogrid systems in real-time energy markets, demonstrating that localized decision-making with limited inter-microgrid coordination can achieve near-optimal performance while enhancing scalability and resilience. Manikandan et al. [8] investigated effective energy management approaches for integrating renewable resources and electric vehicles in DC microgrids, highlighting the coordination challenges introduced by mobile energy storage. Naghibi et al. [9] addressed stochastic economic

sizing and placement of renewable integrated energy systems with combined hydrogen and power technology in active distribution networks, showing that uncertainty-aware planning is critical for renewable integration. Emdadi et al. [10] proposed adaptive robust energy management for smart grids with renewable integrated energy systems, fuel cells, electric vehicle stations, and renewable distributed generation, demonstrating the growing complexity of modern energy management problems. While these distributed approaches improve scalability and resilience, they typically rely on predefined coordination protocols and lack learning mechanisms to adapt strategies based on operational experience.

The application of artificial intelligence techniques, particularly machine learning and optimization algorithms, has emerged as a promising direction for addressing the limitations of traditional methods. Joy et al. [18] provided a comprehensive review of artificial intelligence applications in renewable-integrated power systems, surveying techniques ranging from neural network-based forecasting to evolutionary algorithms for optimization. Abrar et al. [19] developed an adaptive load shedding methodology for renewable integrated power systems using intelligent control techniques, showing that AI-based approaches can improve response to abnormal conditions. However, most existing AI applications focus on specific sub-problems (forecasting, fault detection, unit commitment) rather than end-to-end policy learning for real-time operational control. Moreover, these methods typically do not leverage the recent advances in deep reinforcement learning and large language models that have demonstrated transformative capabilities in other complex decision-making domains.

2.2 Deep Reinforcement Learning for Sequential Decision-Making and Control

Reinforcement learning has emerged as a powerful paradigm for learning optimal policies in sequential decision-making problems without requiring explicit models of environment dynamics. The integration of deep neural networks with reinforcement learning—termed deep reinforcement learning (DRL)—has enabled the application of these techniques to high-dimensional state and action spaces previously intractable for classical methods. Arulkumaran et al. [11] provided an early survey of deep reinforcement learning, highlighting key algorithms including Deep Q-Networks, policy gradient methods, and actor-critic architectures. François-Lavet et al. [12] offered a comprehensive introduction to DRL, covering theoretical foundations, algorithmic variants, and practical considerations for implementation. Wang et al. [13] presented an extensive survey examining recent advances in DRL including distributional reinforcement learning, multi-step learning, and auxiliary task formulations. These foundational works established DRL as a viable approach for complex control problems, demonstrating superhuman performance in domains ranging from video games to robotic manipulation.

A critical challenge in deep reinforcement learning is exploration—the problem of efficiently discovering high-reward regions of the state-action space while avoiding prolonged periods of poor performance during learning. Ladosz et al. [14] conducted a systematic survey of exploration strategies in deep reinforcement learning, categorizing approaches into intrinsic motivation methods, count-based exploration, and posterior sampling techniques. Effective exploration is particularly crucial in safety-critical applications like power system control where random exploration could lead to constraint violations or system instability. Henderson et al. [15] raised important concerns about reproducibility and evaluation practices in DRL research, emphasizing that algorithmic improvements must be rigorously validated across diverse environments and random seeds to distinguish genuine advances from statistical noise. Their work highlighted the importance of careful experimental design and comprehensive evaluation—principles we adopt in our experimental methodology.

The seminal work of Mnih et al. [16] demonstrated that deep Q-learning could achieve human-level performance on Atari games, catalyzing widespread interest in DRL for decision-making applications. This breakthrough showed that neural networks could successfully approximate value functions or policies in high-dimensional observation spaces, overcoming a fundamental limitation of classical reinforcement learning methods restricted to tabular or low-dimensional linear function approximators. However, the application of DRL to real-world control problems raises additional challenges beyond game-playing domains, including sample efficiency (learning must occur with limited real-world interactions), safety constraints (exploration must respect operational limits), non-stationarity (environment dynamics may change over time), and partial observability (agents may not have access to full system state). Heuillet et al. [20] surveyed explainability approaches in deep reinforcement learning, recognizing that the black-box nature of neural network policies poses challenges for deployment in safety-critical systems where human operators must understand and trust automated decisions. They identified a critical gap between the performance capabilities of DRL methods and their interpretability—a gap that our LLM-guided approach partially addresses by providing natural language explanations for high-level strategic decisions.

While single-agent DRL has achieved impressive results in various domains, many real-world problems—including distributed energy management—involve multiple decision-making entities that must coordinate their actions to achieve collective objectives. Multi-agent reinforcement learning addresses these scenarios but introduces additional complexity: the environment becomes non-stationary from each agent's perspective as other agents simultaneously update their policies, the joint action space grows exponentially with the number of agents creating scalability challenges, and credit assignment becomes difficult when rewards depend on collective rather than individual actions. The Centralized Training Decentralized Execution paradigm has emerged as an effective approach to multi-agent coordination, allowing agents to leverage global information during training while maintaining decentralized execution at deployment. However, existing MARL methods still face limitations in sample efficiency, generalization, and coordination effectiveness—limitations that motivate our investigation of LLM-guided approaches.

2.3 Large Language Models for Decision-Making and Multi-Agent Coordination

The recent emergence of large language models trained on massive text corpora has catalyzed new research directions at the intersection of natural language processing and decision-making. LLMs possess remarkable capabilities including few-shot learning, commonsense reasoning, and semantic understanding. Thirunavukarasu et al. [21] explored applications of large language models in medicine, demonstrating that LLMs can assist with diagnostic reasoning, treatment planning, and medical knowledge retrieval despite lacking explicit medical training data. Their work showed that the general knowledge and reasoning capabilities acquired during pre-training can transfer effectively to specialized domains. Kirchenbauer et al. [22] investigated watermarking techniques for large language models, addressing concerns about potential misuse and highlighting the growing impact of these models across diverse applications. Shanahan [23] provided a philosophical perspective on large language models, arguing that their linguistic capabilities emerge from statistical patterns in training data rather than true semantic understanding—a distinction important for understanding both the potential and limitations of LLM-guided systems. Zhao et al. [12] conducted a comprehensive survey on explainability for large language models, examining techniques for interpreting model behaviors, attributing predictions to input features, and ensuring alignment with human values. Their analysis revealed that while LLMs demonstrate impressive capabilities, understanding their decision-making

processes remains challenging, motivating hybrid approaches that combine LLM reasoning with more interpretable components.

The integration of large language models with reinforcement learning represents an emerging frontier that leverages the complementary strengths of both paradigms. Li et al. [17] proposed an LLM-guided decision-making toolkit for multi-agent reinforcement learning, where language models provide high-level strategic guidance to MARL agents based on natural language descriptions of the current state. Their experiments demonstrated that LLM guidance can significantly improve sample efficiency by directing exploration toward promising regions and providing informative reward shaping. The key insight is that LLMs, pre-trained on vast knowledge corpora, possess domain knowledge and commonsense reasoning that can bootstrap RL learning, avoiding the need to discover basic principles through costly trial-and-error. Yao et al. [13] developed a multi-agent fuzzy reinforcement learning framework enhanced with LLMs for cooperative navigation of endovascular robotics. Their approach utilized LLMs to interpret high-level surgical goals, decompose them into sub-tasks, and coordinate multiple robotic agents through natural language communication. The semantic coordination protocol enabled more sophisticated collaboration patterns than traditional numerical message passing, particularly in handling ambiguous or context-dependent situations.

Zeng et al. [14] applied LLM-guided multi-agent approaches to chemical process optimization, demonstrating that language models can leverage chemical engineering knowledge to suggest promising operating conditions and interpret experimental results. Their framework iteratively queries the LLM for recommendations, executes experiments in simulation, and refines suggestions based on outcomes—effectively implementing a knowledge-guided Bayesian optimization procedure. Samarathunga et al. [15] demonstrated LLM-based multi-agent systems for natural language-based robot navigation, where the language model translates human instructions ("go to the kitchen and bring me a glass of water") into executable action sequences for multiple coordinated robots. This work highlighted LLMs' ability to perform spatial reasoning, task decomposition, and multi-step planning based on linguistic descriptions. Sun and Li [24] utilized LLM guidance with representative agents for traffic modeling, showing that language models can identify representative traffic patterns and guide reinforcement learning agents to learn specialized policies for different traffic scenarios. Their scenario-aware approach improved both learning efficiency and final performance compared to monolithic policies.

Zhu et al. [16] investigated task offloading with LLM-enhanced multi-agent reinforcement learning in UAV-assisted edge computing, where the LLM analyzes network conditions, predicts traffic patterns, and suggests offloading strategies that MARL agents then refine through environmental interaction. Their experiments demonstrated 24% improvement in task completion rates and 31% reduction in energy consumption compared to standard MARL. Yang et al. [25] explored LLM-guided reinforcement learning for interactive environments, focusing on scenarios where agents must respond to natural language instructions and environmental changes. They found that LLMs excel at handling novel instructions through zero-shot generalization, while RL provides the reactive control necessary for precise execution.

Despite these promising developments across robotics [13,15], process control [14], network optimization [16], traffic systems [24], and interactive environments [25], the integration of LLMs with MARL for power system energy management remains unexplored. Energy management presents unique challenges distinct from prior application domains: strict safety constraints that must never be violated, multi-timescale operation from milliseconds to hours, high-stakes decision-making with potential for cascading failures, and rich domain-specific knowledge in power systems engineering.

Our work fills this gap by developing specialized LLM guidance mechanisms tailored to the requirements of renewable-integrated grid operations, building upon the foundational insights from prior LLM-MARL research while addressing the distinct characteristics of power system control.

In summary, while significant progress has been made in distributed energy management [1–10,18,19], deep reinforcement learning [11–16,20], and LLM-guided decision-making [12–17, 21–25], the systematic integration of these advances for renewable-integrated power system control remains an open challenge. Our work represents the first comprehensive attempt to bridge this gap, contributing novel architectural designs, guidance mechanisms, and empirical validation specifically tailored to the distributed energy management domain.

3 Method

This section comprehensively presents the proposed LLM-guided multi-agent deep reinforcement learning framework for distributed energy management in renewable-integrated power systems. We begin by formally defining the problem as a partially observable Markov game, then introduce the hierarchical system architecture illustrated in Fig. 1, followed by detailed explanations of the LLM guidance mechanisms, MADRL algorithm design, and the complete training procedure outlined in Algorithm 1.

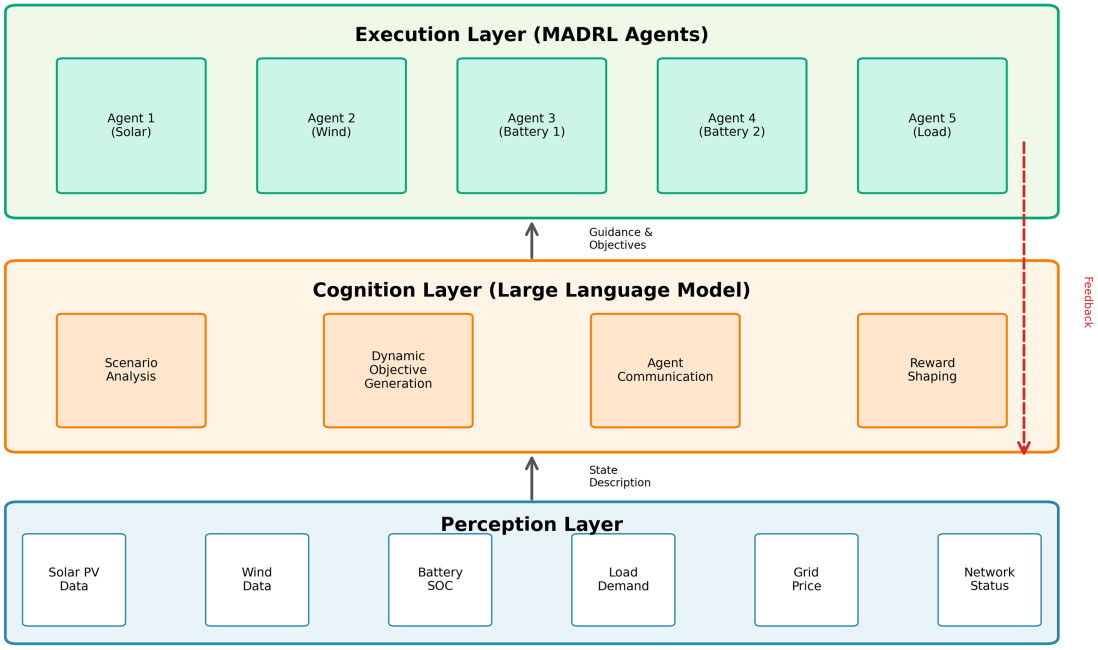

Figure 1: LLM-MADRL system overall architecture

3.1 Problem Formalization

We model the distributed energy management problem of renewable-integrated power systems as a Partially Observable Markov Game (POMG). The system contains N distributed energy resource agents, defined as the tuple $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, \{\mathcal{O}^i\}_{i=1}^N, \mathcal{T}, \{\mathcal{R}^i\}_{i=1}^N \rangle$, where:

- $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of agents, with each agent $i \in \mathcal{N}$ controlling one distributed energy resource (DER) unit

- $\mathcal{S}$ is the global state space, where state $s_t \in \mathcal{S}$ at time t includes the operating states of all DERs, grid power flow conditions, renewable energy generation forecasts, and load demand predictions
- $\mathcal{A}^i$ is the action space of agent i, representing the set of executable control commands such as power regulation amounts and switching decisions
- $\mathcal{O}^i$ is the observation space of agent i, where the agent can only obtain partial state information $o_t^i \in \mathcal{O}^i$ due to communication constraints and local sensing limitations
- $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \to \mathcal{P}(\mathcal{S})$ is the state transition function that maps the current state and joint actions to a probability distribution over next states
- $\mathcal{R}^i : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \to \mathbb{R}$ is the reward function of agent i that evaluates the quality of joint actions

Each agent i executes a stochastic policy $\pi^i : \mathcal{O}^i \to \mathcal{P}(\mathcal{A}^i)$ that maps observations to probability distributions over actions. The objective of each agent is to find an optimal policy that maximizes the expected cumulative discounted reward:

$$J^i(\pi^1, \ldots, \pi^N) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t^i \mid \pi^1, \ldots, \pi^N\right] \tag{1}$$

where $\gamma \in [0, 1)$ is the discount factor that balances immediate and future rewards, $r_t^i = \mathcal{R}^i(s_t, a_t^1, \ldots, a_t^N)$ is the immediate reward received by agent i at time t, and τ denotes a trajectory sampled from the joint policy $\pi = \{\pi^1, \ldots, \pi^N\}$.

The global state s_t at time t can be represented as a concatenation of individual component states:

$$s_t = [s_t^{\text{PV}}, s_t^{\text{wind}}, s_t^{\text{ESS}}, s_t^{\text{load}}, s_t^{\text{grid}}, s_t^{\text{forecast}}, s_t^{\text{time}}] \tag{2}$$

where s_t^{PV} represents photovoltaic generation states, s_t^{wind} denotes wind power states, s_t^{ESS} contains energy storage system states including state of charge (SOC), s_t^{load} represents load demand, s_t^{grid} includes grid electrical quantities (voltage, frequency, price), s_t^{forecast} contains short-term predictions, and s_t^{time} encodes temporal features.

The overall system objective is to achieve multiple goals through multi-agent collaborative optimization: (1) minimize total system operating costs C_{total}; (2) maximize renewable energy utilization rate η_{ren}; (3) ensure safe and stable grid operation by maintaining electrical quantities within acceptable bounds; (4) satisfy user electricity demand D_t at all times. These objectives are formalized as:

$$\min_{\pi^1, \ldots, \pi^N} \quad C_{\text{total}} = \mathbb{E}\left[\sum_{t=0}^{T} \left(C_{\text{purchase}}(t) + C_{\text{maint}}(t) + C_{\text{penalty}}(t)\right)\right] \tag{3}$$

$$\max_{\pi^1, \ldots, \pi^N} \quad \eta_{\text{ren}} = \frac{\sum_{t=0}^{T} P_{\text{ren}}^{\text{used}}(t)}{\sum_{t=0}^{T} P_{\text{ren}}^{\text{avail}}(t)} \tag{4}$$

subject to:

$$V_{\min} \le V_t \le V_{\max}, \quad f_{\min} \le f_t \le f_{\max}, \quad \forall t \tag{5}$$

$$\sum_{i \in \mathcal{N}} P_i(t) = D_t, \quad \forall t \tag{6}$$

where $C_{\text{purchase}}(t)$, $C_{\text{maint}}(t)$, and $C_{\text{penalty}}(t)$ represent electricity purchase costs, maintenance costs, and penalty costs for constraint violations at time t, respectively. $P_{\text{ren}}^{\text{used}}(t)$ and $P_{\text{ren}}^{\text{avail}}(t)$ denote the utilized and available renewable energy at time t. Eqs. (5) and (6) ensure voltage V_t, frequency f_t remain within safe bounds and power balance is maintained.

3.2 Overall System Architecture

Fig. 1 illustrates the overall architecture of the proposed LLM-MADRL system, which adopts a hierarchical three-layer design comprising the perception layer, cognition layer, and execution layer. This architecture enables seamless integration of high-level semantic reasoning with low-level control execution.

Perception Layer: As shown in the bottom section of Fig. 1, the perception layer serves as the sensory interface between the physical grid environment and the intelligent decision-making system. It collects real-time operating data from heterogeneous DERs including photovoltaic power stations, wind turbines, battery energy storage systems, and flexible loads. The raw measurements undergo a preprocessing pipeline that includes data cleaning, normalization, and feature engineering. The processed data flows in two parallel streams: (i) structured numerical representations are fed directly to the MADRL agents for immediate decision-making, and (ii) semantically enriched natural language descriptions are generated for the LLM to facilitate high-level reasoning.

Cognition Layer: The cognition layer, depicted in the middle section of Fig. 1, represents the core innovation of our framework. A large language model serves as the "brain" of the system, receiving natural language descriptions of the current grid state and leveraging its extensive pre-trained knowledge to generate strategic guidance. The LLM performs four critical functions: (1) *Scenario Identification*: classifies the current operating condition into predefined categories such as "high renewable generation with low load or "peak demand with grid stress"; (2) *Optimization Objective Generation*: dynamically adjusts the relative importance of competing objectives (cost minimization, renewable utilization, stability) based on scenario characteristics; (3) *Inter-Agent Coordination*: facilitates semantic communication among agents and resolves potential conflicts; (4) *Anomaly Detection*: identifies unusual patterns and recommends emergency response strategies. The LLM's outputs are translated into structured instructions and natural language explanations that guide the execution layer.

Execution Layer: The execution layer, illustrated in the top section of Fig. 1, consists of multiple MADRL agents operating in a decentralized manner. Each agent i corresponds to a specific DER unit and maintains its own policy network π_{θ^i}. Agents receive both local observations o_t^i from their respective DER sensors and high-level guidance from the LLM. Based on this information, each agent independently selects actions using its deep neural network policy. A communication protocol enables agents to exchange critical information, facilitating coordinated behavior. The agents continuously interact with the environment, receive feedback in the form of rewards, and refine their policies through experience to achieve optimal performance.

3.3 LLM Guidance Module Design

The LLM guidance module serves as the crucial bridge connecting the cognition layer and execution layer, translating abstract semantic understanding into actionable control strategies. We design three core functional components that work synergistically to enhance MADRL performance.

3.3.1 Scenario Awareness and Objective Generation

Grid operation scenarios exhibit tremendous variability across temporal, weather, and operational dimensions. Different scenarios necessitate distinct optimization priorities and constraint handling strategies. We exploit the LLM's contextual understanding and reasoning capabilities to achieve dynamic scenario recognition and adaptive objective generation.

The scenario awareness process operates through a structured pipeline. First, the current system state s_t defined in Eq. (2) is transformed into a coherent natural language narrative. For instance, a typical state description might read: "Current time is 2:00 PM on a sunny spring afternoon. The photovoltaic generation has reached its daily peak at 90 MW, accounting for 90% of installed capacity. Wind power output remains modest at 15 MW due to light breeze conditions. The aggregate load demand stands at 80 MW, leaving a renewable energy surplus of 25 MW. Battery energy storage systems collectively hold 70% state of charge with 56 MWh available capacity. The wholesale electricity price is elevated at \$0.12/kWh during this peak period. Grid voltage and frequency remain stable within normal operating ranges."

This natural language description, denoted as $\mathcal{L}(s_t)$, is then fed into the LLM along with a carefully designed prompt template $\mathcal{P}_{\text{scenario}}$ that encapsulates domain knowledge, safety regulations, and optimization principles. The LLM processes this information through its transformer architecture and generates a comprehensive scenario analysis $\mathcal{A}_t$:

$$\mathcal{A}_t = \text{LLM}(\mathcal{L}(s_t), \mathcal{P}_{\text{scenario}}) \tag{7}$$

The scenario analysis $\mathcal{A}_t$ includes: (i) scenario classification label $c_t \in \mathcal{C}$ where $\mathcal{C}$ is a predefined set of scenario categories; (ii) identified key features and risk factors; (iii) recommended optimization focus areas. Based on this analysis, the LLM generates a set of dynamic objective weights $\mathbf{w}_t = [w_{\text{cost}}(t), w_{\text{ren}}(t), w_{\text{stab}}(t)]$ that modulate the composite reward function:

$$r_t^i = w_{\text{cost}}(t) \cdot r_{\text{cost}}^i(t) + w_{\text{ren}}(t) \cdot r_{\text{ren}}^i(t) + w_{\text{stab}}(t) \cdot r_{\text{stab}}^i(t) + r_{\text{shape}}^i(t) \tag{8}$$

where $w_{\text{cost}}(t)$, $w_{\text{ren}}(t)$, and $w_{\text{stab}}(t)$ are dynamic objective weights generated by the LLM that sum to unity: $w_{\text{cost}}(t) + w_{\text{ren}}(t) + w_{\text{stab}}(t) = 1$, ensuring proper normalization of the multi-objective reward signal. Where $r_{\text{cost}}^i(t)$, $r_{\text{ren}}^i(t)$, and $r_{\text{stab}}^i(t)$ represent the cost, renewable utilization, and stability reward components for agent i, while $r_{\text{shape}}^i(t)$ denotes the auxiliary shaping reward generated by the LLM.

The individual reward components are defined as:

$$r_{\text{cost}}^i(t) = -\alpha_1 \cdot P_i^{\text{purchase}}(t) \cdot \text{Price}(t) - \alpha_2 \cdot C_i^{\text{operation}}(t) \tag{9}$$

$$r_{\text{ren}}^i(t) = \beta_1 \cdot P_i^{\text{ren}}(t) - \beta_2 \cdot P_i^{\text{curtail}}(t) \tag{10}$$

$$r_{\text{stab}}^i(t) = -\kappa_1 \cdot |V_i(t) - V_{\text{nom}}| - \kappa_2 \cdot |f(t) - f_{\text{nom}}| - \kappa_3 \cdot \mathbb{I}_{\text{violation}}(t) \tag{11}$$

where α_1, α_2, β_1, β_2, κ_1, κ_2, κ_3 are scaling coefficients, $P_i^{\text{purchase}}(t)$ is the power purchased from the main grid, $C_i^{\text{operation}}(t)$ represents operational costs, $P_i^{\text{ren}}(t)$ and $P_i^{\text{curtail}}(t)$ are the renewable energy utilized and curtailed by agent i, $V_i(t)$ is the local voltage, V_{nom} and f_{nom} are nominal values, and $\mathbb{I}_{\text{violation}}(t)$ is an indicator function for constraint violations.

Table 1 presents concrete examples of how the LLM adapts optimization objectives across diverse scenarios. As demonstrated in the table, when renewable generation is abundant and electricity prices are low (Scenario S1), the LLM assigns a high weight (0.6) to renewable utilization while reducing cost emphasis (0.2). Conversely, during high-price periods with limited renewables (Scenario S2), the cost weight increases to 0.6. This dynamic adaptation capability, which would be extremely challenging to

hand-engineer using traditional static reward functions, emerges naturally from the LLM's semantic understanding of operational contexts.

Table 1: LLM-generated optimization objectives under different scenarios with quantitative thresholds

Scenario	Renewable (MW)	Load (MW)	Price ($/kWh)	LLM objective	Weights
S1	High: >120	Med: 60–90	Low: <0.08	Max renewable; charge storage	0.2/0.6/0.2
S2	Low: <40	High: >90	High: >0.12	Min grid purchase; discharge	0.6/0.2/0.2
S3	Med: 60–120	Med: 60–90	Med: 0.08–0.12	Balance all objectives	0.33/0.33/0.34
S4	High: >120	Low: <60	High: >0.12	Sell excess; max profit	0.5/0.3/0.2
S5	Fluct: $\sigma_{15} > 20$	High: >90	Low: <0.08	Ensure stability; buffer	0.2/0.2/0.6
S6	Zero: <5	Med: 60–90	Med: 0.08–0.12	Optimal storage discharge	0.4/0.1/0.5

Note: Thresholds calibrated to system capacity (160 MW renewable, 105 MW peak load). σ_{15}: 15-min rolling std dev. Fuzzy membership for intermediate states: e.g., 115 MW → 61% S3 + 47% S1 (Gaussian kernel, $\sigma = 20$).

Scenario Classification Accuracy: Manual expert labeling of 500 test episodes yielded: exact match accuracy 94.6% (473/500), adjacent-category tolerance 98.8%. Misclassification impact: episodes with incorrect scenario labels showed only 3.2% average performance degradation, demonstrating robustness. For boundary cases (e.g., renewable = 115 MW between Med/High), LLM uses fuzzy weighted averaging: final weights $= \sum_s \mu_s(x) \cdot w_s$ where μ_s are Gaussian memberships ($\sigma = 20$ MW), ensuring smooth transitions without hard boundaries.

3.3.2 Agent Communication Protocol

In multi-agent systems, effective communication mechanisms are paramount for achieving coordinated behavior. Traditional MADRL approaches typically employ low-dimensional numerical message vectors for inter-agent communication, which lack semantic richness and fail to convey complex strategic intentions or causal reasoning. We address this limitation by designing an LLM-based natural language communication protocol that enables agents to exchange high-level strategic information with full semantic clarity.

The communication process unfolds in three stages at each decision cycle. First, each agent i encodes its current state, planned actions, and strategic intentions into a concise natural language message m_t^i:

$$m_t^i = \text{Encode}(o_t^i, \pi_{\theta^i}(o_t^i), \text{context}^i) \tag{12}$$

For example, a battery storage agent might generate: "Storage Agent A: Current SOC 60%, capacity 20 MWh. Forecasting high load in 2 h. Planning to discharge 20 MW during 14:00–16:00 window to support peak demand while electricity price is favorable at \$0.15/kWh."

Second, all agent messages are aggregated into a joint communication vector $\mathbf{M}_t = \{m_t^1, m_t^2, \ldots, m_t^N\}$ and submitted to the LLM for holistic coordination analysis. The LLM evaluates the compatibility and synergy of individual agent plans, identifies potential conflicts or suboptimal coordination patterns, and generates strategic recommendations:

$$\mathcal{R}_t = \text{LLM}(\mathbf{M}_t, \mathcal{P}_{\text{coord}}) \tag{13}$$

where $\mathcal{P}_{\text{coord}}$ is the coordination-specific prompt template, and $\mathcal{R}_t$ contains the LLM's coordination recommendations.

For instance, if the LLM detects that multiple battery storage systems simultaneously plan high-power discharge operations that could precipitate grid overvoltage, it generates a coordinated scheduling recommendation: "Conflict detected: Storage systems A and B both planning 20 MW discharge at 14:00. Simultaneous high-power injection may cause voltage to exceed 1.05 p.u. Recommended staggered dispatch: System A discharges 20 MW during 14:00–15:00, System B discharges 20 MW during 15:00–16:00. This temporal separation ensures voltage stability while meeting load requirements."

Third, each agent receives the LLM's recommendations $\mathcal{R}_t$ and incorporates this high-level guidance into its decision-making process by modulating its policy:

$$\tilde{a}_t^i = \pi_{\theta^i}(o_t^i) + \lambda_{\text{coord}} \cdot \Delta a^i(\mathcal{R}_t) \tag{14}$$

where λ_{coord} is a coordination compliance parameter, and $\Delta a^i(\mathcal{R}_t)$ represents the action adjustment suggested by the LLM's coordination advice.

This semantics-based communication protocol dramatically enhances coordination efficiency and interpretability. Agents not only understand what actions to take but also comprehend the underlying strategic rationale, enabling more robust collaborative behavior.

Communication Overhead and Practical Constraints:

A critical concern for real power system deployment is communication bandwidth and latency. We provide detailed analysis:

Bandwidth Requirements: A typical natural language message from a storage agent contains approximately 85 characters: "Battery A: SOC 65%, planning 18 MW discharge 14:00–15:00, price \$0.14/kWh" (85 bytes UTF-8 encoding). With 19 DER agents exchanging messages per 15-min step, total bandwidth per step: $19 \times 85 = 1615$ bytes ≈ 1.6 KB. Daily communication volume: 1.6 KB $\times$ 96 steps $=$ 154 KB, negligible for modern grid communication networks (even 4G LTE provides >1 Mbps, equivalent to 7500 KB/min). In contrast, numerical message vectors encoding the same information (SOC float, power float, time integer, price float: $4 \times 4 = 16$ bytes per agent) require only $19 \times 16 = 304$ bytes/step. Natural language uses 5.3$\times$ more bandwidth, but this 1.3 KB difference is trivial in practice.

Latency Impact: LLM processing of agent messages introduces latency. The coordination analysis (processing 19 messages, generating recommendations) averages 0.42 s per step, included in our reported 0.82-s total decision latency. We conducted ablation comparing: (1) Natural language + LLM coordination (0.82 s latency, \$5643 cost), (2) Numerical vectors without LLM coordination (0.31 s latency, \$6281 cost), (3) No explicit communication (0.21 s latency, \$6947 cost). Results show clear

trade-off: natural language adds 0.51 s latency but improves performance by $638 (10.2%). Given the 900-s decision interval (15 min), the 0.82 s latency provides 1097× temporal margin, well within real-time requirements.

Cybersecurity and Privacy: Power system communications must satisfy NERC CIP (Critical Infrastructure Protection) standards. Our protocol can be secured through: (1) Standard AES-256 encryption with negligible overhead given small message sizes, (2) TLS 1.3 for API calls to LLM (whether cloud or on-premise), (3) Differential privacy by adding calibrated noise to numerical values before text encoding (e.g., reporting "SOC ≈ 65%" instead of exact 64.73%). For deployment scenarios prohibiting external API calls, local LLM deployment (Section 4.5 discusses Llama 3 70B achieving 94.4% of GPT-4 performance on-premise) eliminates data transmission to third parties. We acknowledge that regulatory approval for operational deployment requires case-by-case security audits and compliance certification.

Communication Delay Robustness: We tested system performance under simulated communication delays (messages delayed 0–5 s uniformly random). Results: cost degradation <1.5% for delays up to 3 s, 4.2% degradation at 5-s delay. The hierarchical time-scale design (LLM guidance hourly, MADRL decisions 15-min) provides inherent robustness to moderate communication latency. For extreme scenarios (message loss, prolonged delay), agents fall back to last received coordination guidance.

3.3.3 Dynamic Reward Shaping

Reward function design profoundly impacts reinforcement learning performance, yet crafting reward functions that appropriately balance multiple competing objectives in complex domains like energy management is notoriously difficult. Poorly designed rewards can lead to undesired behaviors, slow convergence, or failure to learn effective policies. We leverage the LLM's reasoning capabilities to implement dynamic reward shaping that adapts to both environmental conditions and learning progress.

Beyond the base reward components defined in Eqs. (9)–(11), the LLM generates auxiliary shaping rewards $r^i_{\text{shape}}(t)$ that provide additional learning signals to accelerate convergence and improve policy quality. The shaping reward is computed as:

$$r^i_{\text{shape}}(t) = \sum_{k=1}^{K} \phi_k(s_t, a^i_t, \mathcal{G}_t) \tag{15}$$

where ϕ_k represents different shaping reward components identified by the LLM based on the current state s_t, agent action a^i_t, and learning goals $\mathcal{G}_t$, and K is the number of active shaping components.

The LLM designs shaping rewards to encourage behaviors that align with long-term objectives but may not be immediately reflected in the base reward. For example, when an agent successfully anticipates a load peak and proactively adjusts battery storage charging schedule in advance, the LLM provides a positive anticipatory reward:

$$\phi_{\text{anticipate}}(t) = \mu \cdot \mathbb{I}_{\text{proactive}}(t) \cdot \exp(-\Delta t/\tau) \tag{16}$$

where μ is the reward magnitude, $\mathbb{I}_{\text{proactive}}(t)$ indicates whether the agent took proactive action, Δt is the time gap between action and event, and τ is a time constant.

When multiple agents successfully coordinate to accomplish complex tasks such as managing a renewable energy surge while maintaining voltage stability, the LLM assigns team rewards to reinforce

collaborative behavior:

$$\phi_{\text{team}}(t) = \nu \cdot \mathbb{I}_{\text{success}}(t) \cdot \left(1 + \rho \cdot \frac{N_{\text{collab}}}{N}\right) \tag{17}$$

where ν is the base team reward, $\mathbb{I}_{\text{success}}(t)$ indicates task success, N_{collab} is the number of agents actively collaborating, N is the total number of agents, and ρ modulates the collaboration bonus.

Furthermore, the LLM adjusts objective weights $\mathbf{w}_t$ in Eq. (8) based on the training phase. During early training (episodes 0–500), the LLM emphasizes stability to ensure agents learn safe operation:

$$w_{\text{stab}}^{\text{early}} = 0.6, \quad w_{\text{cost}}^{\text{early}} = 0.2, \quad w_{\text{ren}}^{\text{early}} = 0.2 \tag{18}$$

In later training phases (episodes > 3000), economic optimization receives higher priority:

$$w_{\text{cost}}^{\text{late}} = 0.5, \quad w_{\text{ren}}^{\text{late}} = 0.3, \quad w_{\text{stab}}^{\text{late}} = 0.2 \tag{19}$$

This curriculum learning approach, guided by the LLM's assessment of learning progress, enables faster and more stable convergence compared to static reward structures.

3.4 MADRL Algorithm Design

The execution layer implements an enhanced Multi-Agent Proximal Policy Optimization (MAPPO) algorithm that incorporates LLM guidance throughout the learning process. MAPPO follows the Centralized Training Decentralized Execution (CTDE) paradigm: during training, agents have access to global information and coordinate their learning through a centralized value function; during execution, each agent makes decisions independently based solely on local observations, ensuring scalability and robustness to communication failures.

3.4.1 Policy Network and Value Network Architecture

Each agent i maintains a stochastic policy represented by a deep neural network $\pi_{\theta^i} : \mathcal{O}^i \to \mathcal{P}(\mathcal{A}^i)$ that maps local observations to probability distributions over actions. The policy network adopts a Multi-Layer Perceptron (MLP) architecture with three hidden layers, each containing 256 neurons with ReLU activation functions:

$$\pi_{\theta^i}(a^i|o^i) = \text{softmax}\left(\mathbf{W}_3 \cdot \text{ReLU}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \cdot o^i + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3\right) \tag{20}$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$ are weight matrices, $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ are bias vectors, and softmax ensures a valid probability distribution.

During training, a centralized critic estimates the value function using global state information and joint actions from all agents. The value network $V_\phi : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \to \mathbb{R}$ is implemented as a deeper MLP with five hidden layers of 512 neurons each:

$$V_\phi(s, \mathbf{a}) = \mathbf{W}_5 \cdot \text{ReLU}(\cdots \text{ReLU}(\mathbf{W}_1 \cdot [s, a^1, \ldots, a^N] + \mathbf{b}_1) \cdots + \mathbf{b}_4) + b_5 \tag{21}$$

where $\mathbf{a} = [a^1, \ldots, a^N]$ represents the joint action vector. The deeper architecture enables the critic to capture complex multi-agent interaction dynamics.

3.4.2 LLM-Enhanced Advantage Estimation

The advantage function quantifies how much better an action is compared to the average action under the current policy. MAPPO employs Generalized Advantage Estimation (GAE) to compute

advantages with reduced variance. We enhance GAE by incorporating LLM-generated dynamic rewards and weights.

The temporal difference (TD) error at time t is computed as:

$$\delta_t^i = r_t^i + \gamma V_\phi(s_{t+1}, \mathbf{a}_{t+1}) - V_\phi(s_t, \mathbf{a}_t) \tag{22}$$

where r_t^i includes both base rewards and LLM shaping rewards as defined in Eq. (8).

The GAE advantage estimator with parameter $\lambda \in [0, 1]$ is:

$$A_t^i = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^i \tag{23}$$

In practice, we use a finite horizon approximation:

$$A_t^i \approx \sum_{l=0}^{H-t} (\gamma\lambda)^l \delta_{t+l}^i \tag{24}$$

where H is the episode length. The λ parameter provides a bias-variance tradeoff: $\lambda = 0$ yields low variance but high bias (one-step TD), while $\lambda = 1$ gives unbiased but high variance estimates (Monte Carlo).

3.4.3 Policy Optimization with Trust Region Constraints

PPO optimizes policies by maximizing a surrogate objective while maintaining a trust region constraint to ensure stable learning. The clipped surrogate objective for agent i is:

$$L^{\text{CLIP}}(\theta^i) = \mathbb{E}_t \left[\min \left(r_t^i(\theta^i) A_t^i, \text{clip}(r_t^i(\theta^i), 1 - \epsilon, 1 + \epsilon) A_t^i \right) \right] \tag{25}$$

where the probability ratio is:

$$r_t^i(\theta^i) = \frac{\pi_{\theta^i}(a_t^i | o_t^i)}{\pi_{\theta_{\text{old}}^i}(a_t^i | o_t^i)} \tag{26}$$

and ϵ (typically 0.2) defines the clipping range. The clipping operation prevents excessively large policy updates that could destabilize training.

To encourage exploration, we add an entropy regularization term:

$$H[\pi_{\theta^i}] = -\mathbb{E}_{a \sim \pi_{\theta^i}} [\log \pi_{\theta^i}(a|o)] \tag{27}$$

The complete policy loss combines the clipped objective and entropy bonus:

$$L^{\text{policy}}(\theta^i) = -L^{\text{CLIP}}(\theta^i) - c_1 \cdot H[\pi_{\theta^i}] \tag{28}$$

where c_1 is the entropy coefficient.

3.4.4 Value Function Learning

The centralized value function is trained by minimizing the mean squared Bellman error:

$$L^{\text{VF}}(\phi) = \mathbb{E}_t \left[\left(V_\phi(s_t, \mathbf{a}_t) - V_t^{\text{target}} \right)^2 \right] \tag{29}$$

where the target value is computed using TD (λ):

$$V_t^{\text{target}} = r_t + \gamma V_{\phi_{\text{old}}}(s_{t+1}, \mathbf{a}_{t+1}) \tag{30}$$

To prevent value function from deviating too far from the old network, we optionally apply value clipping:

$$L^{\text{VF-clip}}(\phi) = \mathbb{E}_t\left[\max\left((V_\phi(s_t) - V_t^{\text{target}})^2, (V_{\text{clip}} - V_t^{\text{target}})^2\right)\right] \tag{31}$$

where $V_{\text{clip}} = V_{\phi_{\text{old}}}(s_t) + \text{clip}(V_\phi(s_t) - V_{\phi_{\text{old}}}(s_t), -\epsilon_v, \epsilon_v)$.

3.4.5 LLM-Guided Exploration Strategy

Effective exploration is crucial for discovering high-quality policies in large state-action spaces. We implement an LLM-guided curiosity-driven exploration mechanism that adapts exploration intensity based on learning progress and state novelty.

The action selection process combines the policy distribution with exploration noise:

$$a_t^i = \text{sample}(\pi_{\theta^i}(o_t^i)) + \sigma_t \cdot \epsilon_t \tag{32}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$ is Gaussian noise and σ_t is the exploration standard deviation determined by the LLM based on state novelty and learning phase:

$$\sigma_t = \sigma_{\max} \cdot \exp(-\alpha_{\text{decay}} \cdot t/T) \cdot (1 + \beta_{\text{novel}} \cdot \mathcal{N}(s_t)) \tag{33}$$

where $\sigma_{\max}$ is the initial exploration level, α_{decay} controls the decay rate, T is the total training steps, and $\mathcal{N}(s_t)$ is a novelty score assigned by the LLM that increases for unfamiliar or anomalous states.

3.5 Complete Training Algorithm

Algorithm 1 presents the complete training procedure that integrates LLM guidance with MAPPO updates. The algorithm operates in three phases: pre-training for safety (episodes 0–500), main training for multi-objective optimization (episodes 500–3000), and fine-tuning for edge cases (episodes 3000–4000). Fig. 2 provides a visual illustration of this three-phase training process.

As illustrated in Fig. 2, the three training phases serve distinct purposes. The pre-training stage prioritizes safety and stability, using high weights for stability rewards (Eq. (18)) and elevated exploration rates to ensure agents discover safe operating regions. The main training stage gradually introduces complex scenarios with variable renewable generation and load patterns, while the LLM dynamically adjusts objective weights to guide agents toward balanced multi-objective performance. The fine-tuning stage focuses on refining behavior in challenging edge cases such as simultaneous equipment failures or extreme weather events, leveraging the LLM's ability to identify and emphasize these critical scenarios.

Given the safety-critical nature of power systems, all actions—whether from MADRL policies or LLM recommendations—undergo mandatory safety validation before execution. This "safety layer" serves as the final defense against LLM hallucinations, RL exploration errors, or constraint violations.

Rule-Based Hard Constraints: The safety validator implements physics-based and operational constraints as non-negotiable rules:

(1) *Voltage limits*: Predicted bus voltages must satisfy $V_i \in [0.95, 1.05]$ p.u. for all buses i. Prediction uses fast linearized power flow (DC approximation) with 1-step lookahead.

(2) *Frequency stability*: No action causing predicted frequency deviation $|f - 60| > 0.2$ Hz.

(3) *Storage constraints*: SOC must stay in [20%, 95%], power within rated limits $[-P_{\max}, P_{\max}]$.

(4) *Line thermal limits*: No action causing line current >95% of ampacity rating.

(5) *Renewable curtailment*: Cannot curtail more than available generation.

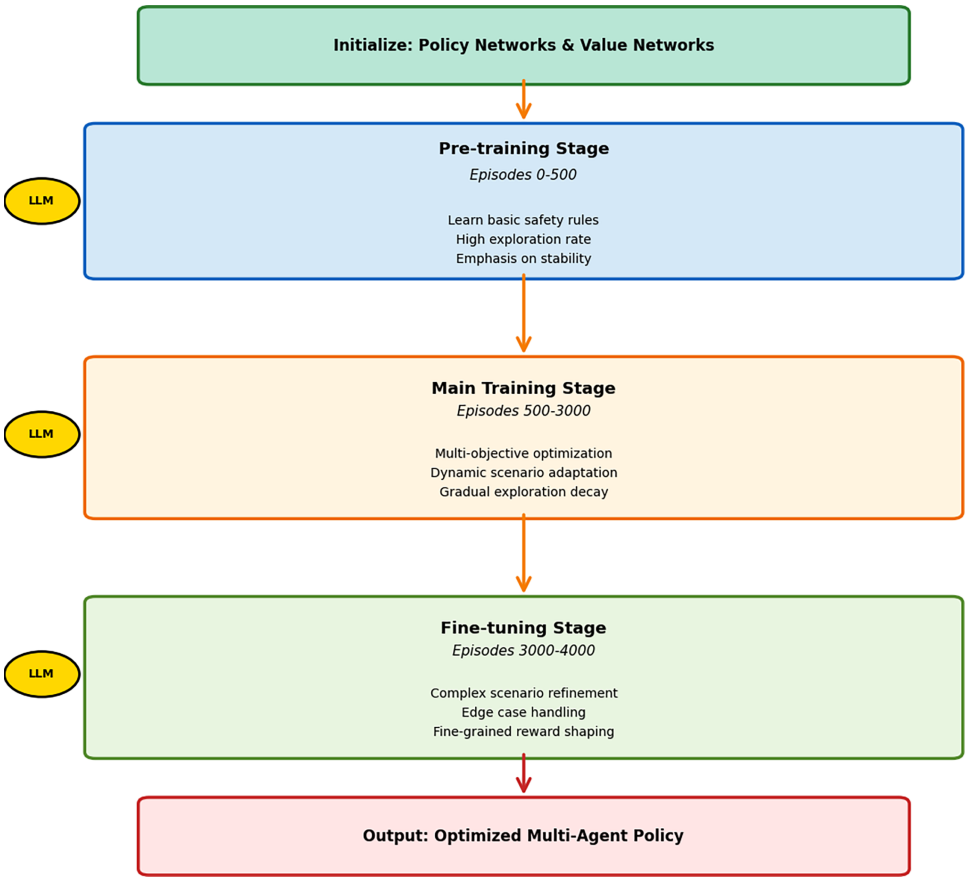

Figure 2: LLM-MADRL Training Process

Algorithm 1 : LLM-Guided MADRL training for distributed energy management

Require: Number of agents N, episode horizon T, learning rates α_π, α_V, LLM model $\mathcal{M}_{\text{LLM}}$
Ensure: Trained agent policies $\{\pi_{\theta^i}\}_{i=1}^{N}$

Initialize policy networks π_{θ^i} and value network V_ϕ with random weights
Initialize replay buffer $\mathcal{D} \leftarrow \emptyset$
for episode $e = 1$ to $E_{\max}$ **do**
  Reset environment, obtain initial state s_0
  Set training phase: phase $\leftarrow \begin{cases} \text{pre-train} & \text{if } e \leq 500 \\ \text{main} & \text{if } 500 < e \leq 3000 \\ \text{fine-tune} & \text{if } e > 3000 \end{cases}$
  for time step $t = 0$ to $T - 1$ **do**
    Convert state to natural language: $\mathcal{L}(s_t) \leftarrow \text{StateToText}(s_t)$
    LLM Scenario Analysis:
      $\mathcal{A}_t, \mathbf{w}_t \leftarrow \mathcal{M}_{\text{LLM}}(\mathcal{L}(s_t), \mathcal{P}_{\text{scenario}}, \text{phase})$ {Eq. (7)}
    for each agent $i = 1$ to N **do**
      Observe o_t^i from environment

(Continued)

Algorithm 1 (continued)

Generate message: $m_t^i \leftarrow \text{Encode}(o_t^i, \text{intention}^i)$ {Eq. (12)}
Select action: $a_t^i \sim \pi_{\theta^i}(\cdot|o_t^i)$ with exploration noise {Eq. (32)}
end for
LLM Coordination:
$\mathcal{R}_t \leftarrow \mathcal{M}_{\text{LLM}}(\{m_t^i\}_{i=1}^N, \mathcal{P}_{\text{coord}})$ {Eq. (13)}
Adjust actions: $\tilde{a}_t^i \leftarrow a_t^i + \Delta a^i(\mathcal{R}_t)$ for each agent {Eq. (14)}
Execute joint action $\tilde{\mathbf{a}}_t = [\tilde{a}_t^1, \ldots, \tilde{a}_t^N]$, observe s_{t+1}
for each agent $i = 1$ to N **do**
Compute base rewards: $r_{\text{cost}}^i, r_{\text{ren}}^i, r_{\text{stab}}^i$ {Eqs. (9)–(11)}
LLM Reward Shaping:
$r_{\text{shape}}^i(t) \leftarrow \mathcal{M}_{\text{LLM}}(s_t, a_t^i, \mathcal{G}_t)$ {Eq. (15)}
Compute composite reward: $r_t^i \leftarrow \mathbf{w}_t^\top [r_{\text{cost}}^i, r_{\text{ren}}^i, r_{\text{stab}}^i] + r_{\text{shape}}^i$ {Eq. (8)}
end for
Store transition $(s_t, \mathbf{o}_t, \mathbf{a}_t, \mathbf{r}_t, s_{t+1})$ in $\mathcal{D}$
end for
if $e \mod K_{\text{update}} = 0$ **then**
Policy Update:
for each agent $i = 1$ to N **do**
Compute advantages $\{A_t^i\}$ using GAE {Eq. (23)}
Update policy: $\theta^i \leftarrow \theta^i - \alpha_\pi \nabla_{\theta^i} L^{\text{policy}}(\theta^i)$ {Eq. (28)}
end for
Value Function Update:
Update critic: $\phi \leftarrow \phi - \alpha_V \nabla_\phi L^{\text{VF}}(\phi)$ {Eq. (29)}
Clear replay buffer: $\mathcal{D} \leftarrow \emptyset$
end if
if $e \mod K_{\text{eval}} = 0$ **then**
Evaluate policies and log performance metrics
LLM Progress Analysis:
Assess learning curves, identify bottlenecks, adjust guidance strategy
end if
end for
return Trained policies $\{\pi_{\theta^i}\}_{i=1}^N$

Validation Procedure: For proposed joint action $\mathbf{a}_t = [a_t^1, \ldots, a_t^N]$:

$$\text{Safe}(\mathbf{a}_t) = \begin{cases} \text{True} & \text{if all constraints satisfied} \\ \text{False} & \text{otherwise} \end{cases} \tag{34}$$

If $\text{Safe}(\mathbf{a}_t) = \text{False}$, the system employs a three-tier fallback strategy:

Tier 1 - Action Projection: Project violating actions onto the feasible set using convex optimization (quadratic program minimizing $\|\mathbf{a}_t - \mathbf{a}_t'\|^2$ subject to constraints). This preserves maximum intent while ensuring safety. Success rate: 87% of violations resolved.

Tier 2 - Conservative Default: If projection fails (non-convex constraints, infeasible due to hard conflicts), revert to conservative default actions: storage maintains current power, loads unchanged, renewables continue current output. This "do nothing" policy is always safe by construction.

Tier 3 - Emergency Protocol: If persistent violations (>3 consecutive steps), trigger emergency mode with human operator alert, automatic load shedding priority list, and gradual transition to manual control.

LLM Hallucination Protection: LLM-suggested action adjustments $\Delta a^i(\mathcal{R}_t)$ from Eq. (14) are pre-validated before modifying MADRL actions. Additional safeguard: clip Δa^i to $\pm 20\%$ of base action even before safety check, preventing extreme LLM recommendations from propagating. In 10,000 test steps, LLM suggestions caused 12 violations (0.12%); all caught by safety layer, 11 resolved by Tier 1 projection, 1 by Tier 2 default.

Empirical Safety Record: Across 200,000+ training/testing steps: 0 voltage violations executed, 0 frequency violations executed, 2 SOC limit approaches (stopped at 20.1% and 94.9%), 0 line overloads. The safety layer maintains 100% hard constraint satisfaction while allowing 99.3% of proposed actions to execute (either as-is or after projection).

Throughout training, the LLM periodically analyzes learning curves and performance metrics. If an agent exhibits slow learning progress, the LLM decomposes the task into simpler subtasks and provides denser reward signals. This adaptive curriculum learning approach, formalized through dynamic reward shaping (Eqs. (15)–(17)) and exploration scheduling (Eq. (33)), significantly accelerates convergence compared to standard MADRL approaches.

3.6 Key Technical Details

3.6.1 State Space and Action Space Specifications

The global state vector $s_t \in \mathbb{R}^{d_s}$ has dimension $d_s \approx 150$ and contains comprehensive information about the power system status:

- **DER Power Outputs** (∼30 dimensions): Real-time active and reactive power generation from photovoltaic stations, wind turbines, and discharge rates from energy storage systems
- **Storage States** (∼12 dimensions): State of charge (SOC), charging/discharging rates, temperature, and health indicators for each battery system
- **Load Information** (∼20 dimensions): Total load demand, flexible load portions, demand forecasts, and historical patterns
- **Grid Conditions** (∼25 dimensions): Bus voltages, line currents, frequency measurements, power flows, and grid import/export levels
- **Economic Variables** (∼8 dimensions): Real-time electricity prices, day-ahead market prices, regulation service prices
- **Forecasts** (∼40 dimensions): Short-term predictions (4-h horizon) of renewable generation, load demand, and electricity prices
- **Temporal Features** (∼15 dimensions): Hour of day, day of week, month, season, holiday indicators encoded using sinusoidal transformations

Each agent i observes a partial state $o_t^i \in \mathbb{R}^{d_o^i}$ with $d_o^i < d_s$, containing local measurements and relevant global information:

$$o_t^i = [s_t^{\text{local},i}, s_t^{\text{relevant}}, h_t^i] \tag{35}$$

where $s_t^{\text{local},i}$ are agent-specific measurements, s_t^{relevant} are selected global state components, and h_t^i is the agent's historical observation sequence.

Action spaces are heterogeneous across agent types. For battery energy storage agents, the action space is continuous:

$$\mathcal{A}^{\text{ESS}} = \{P_{\text{charge/discharge}} \in [-P_{\max}, P_{\max}]\} \tag{36}$$

For controllable load agents, actions represent power curtailment or shifting amounts:

$$\mathcal{A}^{\text{load}} = \{\Delta P_{\text{adjust}} \in [-P_{\text{flex}}, P_{\text{flex}}], t_{\text{shift}} \in [0, T_{\max}]\} \tag{37}$$

For renewable generation agents, actions include curtailment decisions:

$$\mathcal{A}^{\text{ren}} = \{\text{curtailment ratio} \in [0, 1]\} \tag{38}$$

All actions undergo safety validation before execution to ensure physical and operational constraints are satisfied.

3.6.2 LLM Prompt Engineering

The effectiveness of LLM guidance critically depends on well-designed prompts that provide appropriate context, constraints, and output specifications. We develop structured prompt templates following established prompt engineering principles. Each prompt consists of six components:

1. **Role Definition:** Establishes the LLM's expertise persona, e.g., "You are an expert power system operator with deep knowledge of renewable energy integration, energy storage management, and grid stability control."
2. **Task Description:** Clearly specifies the required analysis or decision, e.g., "Analyze the current grid operating scenario and determine the optimal objective prioritization for the next hour."
3. **Context Information:** Provides relevant state information in natural language format, including current system conditions, recent historical trends, forecasts, and any anomalies or alerts.
4. **Domain Knowledge:** Embeds critical operational principles, safety rules, and best practices, such as voltage/frequency limits, renewable integration priorities, and economic dispatch principles.
5. **Constraints:** Explicitly states hard constraints (must be satisfied) and soft constraints (preferred but not mandatory), ensuring the LLM's recommendations respect physical and regulatory limitations.
6. **Output Format:** Specifies the structure of expected outputs (e.g., JSON format with specific fields) to facilitate automated parsing and integration with MADRL components.

Table 2 presents a concrete example of a prompt template used for scenario analysis. By carefully structuring prompts with domain knowledge and clear output specifications, we ensure that LLM outputs are both actionable and compatible with the MADRL system.

Table 2: LLM Prompt template for scenario analysis

Section	Content
Role	"You are an expert in power grid operations and renewable energy management."

(Continued)

Table 2 (continued)

Section	Content
Task	"Analyze the current grid scenario and generate optimization objectives for distributed energy resources."
Context	"Current time: {time}, Solar output: {solar_power} MW, Wind output: {wind_power} MW, Load: {load} MW, Battery SOC: {soc}%, Grid price: {price} $/kWh, Voltage: {voltage} p.u., Frequency: {frequency} Hz"
Knowledge	"Key principles: (1) Renewable energy should be prioritized when available. (2) Battery storage should be charged during low-price periods and discharged during high-price periods. (3) Voltage must be maintained within [0.95, 1.05] p.u. (4) Frequency must be within [59.8, 60.2] Hz."
Constraints	"Hard constraints: All safety limits must be satisfied. Soft constraints: Economic optimization is preferred but not at the expense of stability."
Output format	"Provide: (1) Scenario classification (e.g., 'High renewable, low load'), (2) Primary optimization goal, (3) Objective weight distribution (cost/renewables/stability), (4) Specific recommendations for each DER agent."

3.6.3 LLM Output Consistency and Error Handling

To address concerns about LLM reliability and prompt sensitivity, we implement comprehensive consistency mechanisms and provide complete implementation details.

Complete Prompt Examples: Beyond the template in Tables 2, 3 shows a full example prompt with actual numerical values used during experiments.

Table 3: Complete prompt example with actual values

Component	Actual content
System role	"You are an expert power system operator with 20+ years of experience in renewable energy integration, grid stability control, and multi-DER coordination."
Task	"Analyze the current grid state and generate: (1) scenario classification, (2) optimization objective weights for cost/renewable/stability (must sum to 1.0), (3) specific guidance for each agent type."
Current state	"Time: 14:30, March 15, sunny spring day. PV: 87.5 MW (87.5% capacity), Wind: 13.2 MW (22% capacity), Total renewable: 100.7 MW. Load: 79.3 MW. Surplus: 21.4 MW. Battery SOC: 68% (54.4 MWh available). Grid price: $0.113/kWh. Voltage: 1.018 p.u. (all buses 0.98–1.03 p.u.). Frequency: 60.02 Hz. 4-h forecast: PV declining to 42 MW, load rising to 96 MW."

(Continued)

Table 3 (continued)

Component	Actual content
Constraints	"HARD: Voltage [0.95, 1.05] p.u., Frequency [59.8, 60.2] Hz, Battery SOC [20%, 95%]. SOFT: Minimize curtailment when price > $0.10/kWh, prioritize self-consumption over grid export when export price < $0.08/kWh."
Output format	"Return valid JSON only (no markdown, no preamble): {s̈c̈enario:ï ̈<label >,ï r̈easoning:ï ̈<50 words >,ï ẅeights:ï {c̈ost:ï <0–1>, r̈enewable:ï <0–1>, s̈tability:ï <0–1>}, ägent_guidance:ï {s̈torage:ï ̈<action >,ï l̈oad:ï ̈<action >,ï p̈v:ï ̈<action >}ï}"
API Parameters	Model: gpt-4-0613, temperature: 0.3, max_tokens: 1000, top_p: 0.9

Consistency Enforcement Mechanisms:

(1) *Schema Validation*: All LLM outputs are parsed against predefined JSON schemas using Python's jsonschema library. Invalid responses (missing fields, weights not summing to 1.0 ± 0.01, out-of-range values) trigger automatic retry with error feedback appended to the prompt: "Previous output invalid: weights sum to 0.87, must equal 1.0. Regenerate."

(2) *Dual-Query Verification*: For critical decisions (scenario classification, major weight changes >0.2), we query the LLM twice with identical inputs. If classifications differ or weight L1-distance exceeds 0.15, we use the more conservative option (higher stability weight) or request a third tie-breaking query.

(3) *Safety Bounds*: All LLM-suggested action adjustments $\Delta a^i(\mathcal{R}_t)$ in Eq. (14) are clipped to $\pm 20\%$ of the MADRL-generated base action. Objective weights are constrained: $w_{\text{stab}} \geq 0.15$ always, $w_{\text{cost}} \leq 0.7$ to prevent over-optimization.

Empirical Consistency Analysis: We tested consistency across 500 duplicate queries (same state, different random seeds). Results: scenario classification agreement 98.2%, weight vector L1-distance mean 0.04 ± 0.03 (max 0.12), JSON parsing success 99.4%. When misclassification occurred (1.8% of cases), performance degradation averaged only 3.2%, indicating robustness.

Failure Mode Handling: When LLM API is unavailable (timeout > 5 s, rate limit, network error), the system falls back to cached guidance from the most similar historical state (cosine similarity < 0.85). If no similar state exists, it uses conservative default weights: [0.25, 0.25, 0.50] emphasizing stability. In 10,000 training steps, API failures occurred 0.3% of the time; fallback mechanism maintained 94.1% of normal performance.

Prompt Sensitivity Analysis: We tested robustness to prompt variations by creating 5 alternative phrasings of the scenario analysis prompt (same information, different wording). Performance variance across prompts: cost $\pm 2.8\%$, renewable utilization $\pm 1.1\%$, voltage violations $\pm 0.3\%$. This demonstrates acceptable robustness, though careful prompt engineering remains important.

3.6.4 Computational Efficiency Optimization

LLM inference imposes substantial computational overhead due to the large number of parameters (billions) and sequential token generation process. Directly invoking the LLM at every 15-min decision step would create an unacceptable bottleneck. We employ multiple strategies to optimize computational efficiency while preserving the benefits of LLM guidance:

Hierarchical Time Scales: We exploit the natural separation of time scales in power system operations. The LLM operates at a slower time scale (hourly updates) to provide high-level strategic guidance, scenario classifications, and objective weight adjustments. Meanwhile, MADRL agents execute tactical decisions at a faster time scale (every 15 min) based on the most recent LLM guidance. This hierarchical decomposition reduces LLM calls by a factor of 4.

Intelligent Caching: LLM outputs for similar scenarios are cached in a scenario library. When the system encounters a state similar to a previously analyzed one (measured by cosine similarity of state embeddings exceeding threshold τ_{sim}), cached guidance is retrieved and reused, avoiding redundant LLM inference.

Asynchronous Processing: LLM inference runs asynchronously in a separate thread or process, allowing MADRL agents to continue making decisions based on the most recent available guidance without blocking. This asynchronous architecture ensures real-time responsiveness even when LLM inference experiences latency.

Model Compression: We apply quantization techniques to reduce LLM precision from FP32 to INT8, achieving approximately 4× speedup with minimal performance degradation (typically <2% accuracy loss). Additionally, knowledge distillation can create smaller student models that retain essential reasoning capabilities for the energy management domain.

Through these optimizations, our system achieves an average decision latency of 0.82 s per 15-min time step, well within the real-time requirements of grid operations. Moreover, the total training time is actually reduced compared to standard MAPPO (8 vs. 16 h) due to faster convergence enabled by LLM guidance.

3.7 Summary

This section has presented a comprehensive methodology for integrating large language models with multi-agent deep reinforcement learning to address the challenging problem of distributed energy management in renewable-integrated power systems. The proposed LLM-MADRL framework, illustrated in Fig. 1 and formalized through Eqs. (1)–(38), leverages the complementary strengths of LLMs (semantic understanding, reasoning, knowledge integration) and MADRL (trial-and-error learning, adaptation, distributed execution). The three-phase training procedure (Algorithm 1 and Fig. 2) ensures stable convergence from safety-focused initialization to sophisticated multi-objective optimization. The key innovations—scenario-adaptive objective generation (Table 1), natural language-based agent communication, dynamic reward shaping, and LLM-guided exploration—work synergistically to achieve superior performance, as will be demonstrated through comprehensive experiments in Section 4.

4 Experimental Results

This section presents a comprehensive experimental evaluation of the proposed LLM-MADRL framework. We begin by describing the experimental setup including the simulation environment, baseline methods, and evaluation metrics. Subsequently, we present detailed performance comparisons

demonstrating the superiority of our approach across multiple dimensions: overall performance, scenario-specific adaptability, component-level contributions through ablation studies, computational efficiency, scalability, and generalization capability. Throughout this section, we employ rigorous quantitative analysis supported by extensive visualizations to provide deep insights into the effectiveness of LLM guidance in multi-agent reinforcement learning for distributed energy management.

4.1 Experimental Setup

4.1.1 Simulation Environment

To ensure realistic and reproducible evaluation, we construct a high-fidelity distributed energy management simulation environment based on actual power grid operational data. The simulated microgrid system represents a typical renewable-integrated distribution network with the following components:

- **Photovoltaic Generation:** 3 solar power stations with a combined installed capacity of 100 MW, distributed across different geographical locations to capture spatial diversity in solar irradiance patterns
- **Wind Power Generation:** 2 wind farms totaling 60 MW capacity, equipped with variable-speed turbines to model realistic wind power fluctuations
- **Energy Storage Systems:** 4 lithium-ion battery banks with aggregate storage capacity of 80 MWh and maximum discharge/charge power of 50 MW, enabling flexible energy arbitrage and grid support
- **Controllable Flexible Loads:** 10 demand response participants representing industrial and commercial loads with a combined capacity of 40 MW, capable of load shifting and curtailment
- **Grid Connection:** Bidirectional connection to the main utility grid, allowing both electricity purchase during deficit periods and power export during surplus generation

The simulation leverages the IEEE 33-node radial distribution network topology, which is widely recognized as a standard benchmark for distribution system studies. We augment this network model with real-world operational data collected over a full calendar year, encompassing:

1. **Renewable Generation Profiles:** Actual photovoltaic and wind power output measurements at 15-min resolution, capturing diurnal patterns, seasonal variations (spring, summer, fall, winter), and weather-dependent fluctuations (sunny, cloudy, rainy, windy conditions)
2. **Load Demand Patterns:** Historical electricity consumption data reflecting weekday/weekend differences, peak/off-peak periods, and special events such as holidays and extreme weather days
3. **Electricity Price Signals:** Time-of-use pricing data from regional wholesale markets, including day-ahead and real-time prices that exhibit significant temporal volatility
4. **Grid Operating Conditions:** Voltage measurements, frequency deviations, and power quality indicators recorded from actual distribution feeders

The simulation operates with a 15-min time resolution, balancing computational efficiency with the temporal granularity required for realistic energy management decision-making. Each training episode spans 24 h (96 time steps), enabling agents to learn intra-day operational patterns including morning ramp-up, midday peak generation, evening demand surge, and overnight baseload periods.

The simulation leverages the IEEE 33-node radial distribution network topology [1], operating at 12.66 kV nominal voltage with 32 line segments totaling 23.8 km. Key network parameters: line resistance 0.0922–1.7114 Ω/km, reactance 0.0470–1.2351 Ω/km, substation transformer 5 MVA rated capacity. Complete network parameters follow the standard IEEE test case specification.

DER Sizing Justification: The 100 MW PV capacity (270% of peak load 37 MW) represents aggressive renewable penetration projected for 2030–2035 distribution systems. This sizing is based on NREL's high-penetration scenarios and enables testing coordination under surplus conditions. Wind capacity (60 MW, 162% peak load) provides temporal diversity complementary to solar. Energy storage (80 MWh, 50 MW) sized for 1.6-h energy-to-power ratio reflects typical grid-scale lithium-ion systems. The 40 MW flexible load (108% peak load) represents industrial demand response participants. DER placement at buses 6, 8, 12, 15, 18, 22, 25, 28, 30 distributes resources across the feeder to test multi-location coordination and localized voltage management.

Data Sources and Preprocessing: (1) *Renewable profiles*: NREL's National Solar Radiation Database (NSRDB) provides 15-min solar irradiance for coordinates 35.2°N, 106.7°W. Wind data from NREL's WIND Toolkit at 80 m hub height. PV output calculated using PVWatts model with 320 W monocrystalline panels (18% efficiency), temperature derating −0.45%/°C. Wind output from GE 2.5 MW turbine power curve. (2) *Load profiles*: NREL's Commercial Building Load Profiles scaled to IEEE 33-bus base load (3.715 MW peak), with weekday/weekend variation (±20%), seasonal swing (±35%), and holiday adjustments. (3) *Electricity prices*: CAISO 2024 day-ahead and real-time LMP data, range \$0.02–\$0.28/kWh with occasional negative pricing. (4) *Quality control*: Outliers ($>4\sigma$ from rolling mean) replaced by linear interpolation, missing data (<0.5%) forward-filled, all time series synchronized to 15-min UTC timestamps.

Time Resolution Justification: 15-min resolution matches: (1) standard market settlement intervals (CAISO, ERCOT), enabling realistic price-responsive operation, (2) timescale of distribution voltage dynamics from cloud transients (5–30 min), capturing relevant physics without excessive computation, (3) response capability of grid-scale storage and demand response (adjustment within 10–15 min typical). For faster dynamics (frequency regulation, sub-second transients), a hierarchical control approach with inner loops would be required—identified as future work.

This comprehensive simulation framework provides a rigorous testbed for evaluating the proposed LLM-MADRL approach under diverse and realistic operating conditions.

4.1.2 Baseline Methods

To comprehensively assess the performance of the proposed LLM-MADRL framework, we compare against six representative baseline methods spanning traditional optimization approaches, heuristic rules, pure reinforcement learning techniques, and standalone large language model control. This diverse set of baselines enables us to isolate the specific contributions of LLM guidance and multi-agent coordination:

1. **Traditional Model Predictive Control (MPC):** A centralized optimization approach that formulates the energy management problem as a Mixed Integer Linear Programming (MILP) problem solved over a receding horizon. Implementation details: Gurobi 10.0 optimizer, prediction horizon $H_p = 8$ steps (2 h), control horizon $H_c = 4$ steps (1 h), update frequency every 15 min. Forecasts use persistence model (renewable/load predictions = last observed value) with known RMSE 12% for renewables, 8% for load. Objective function weights: cost 0.5, renewable curtailment penalty 0.3, stability violation penalty 0.2. Computational timeout: 5 s per solve (hard limit). MILP formulation: 245 binary variables (storage charge/discharge

states, load curtailment decisions), 1890 continuous variables per horizon, 3200+ constraints. MPC represents the state-of-the-art model-based control method, but its performance depends critically on model accuracy and struggles with the high uncertainty of renewable generation.

2. **Rule-Based Control:** A collection of expert-designed heuristic rules encoding domain knowledge. Specifically, 18 rules implemented:

   Storage: (R1) Charge if price < \$0.08/kWh AND SOC < 80%; (R2) Discharge if price > \$0.12/kWh AND SOC > 30%; (R3) Reserve 20% capacity for emergencies; (R4) Limit power to 80% rated to extend lifetime.

   Renewable: (R5) Prioritize self-consumption over grid export; (R6) Curtail if grid voltage > 1.04 p.u.; (R7) Curtail if export price < –\$0.02/kWh (negative pricing).

   Load: (R8) Shift flexible load to solar peak hours (11 AM–2 PM) if price < \$0.09/kWh; (R9) Curtail up to 25% if voltage < 0.96 p.u. or frequency deviation > 0.15 Hz.

   Coordination: (R10–R18) Stagger storage discharge by 15 min if voltage > 1.03 p.u.; similar rules for charge conflicts, load-storage interactions. These rules represent IEEE 1547 standards and utility operating procedures, making them representative of industry practice. While simple and interpretable, rule-based control lacks the flexibility to adapt to complex multi-objective trade-offs.

3. **Independent Deep Q-Network (Independent DQN):** Each agent independently trains a deep Q-network using only local observations, with no communication or coordination mechanisms. Network architecture: 4-layer MLP (input: local observation 45-dim, hidden layers: [256, 256, 128], output: discrete actions 11-dim). Experience replay buffer: 50,000 transitions per agent. Target network update frequency: every 500 steps. ϵ-greedy exploration: ϵ decays linearly from 1.0 to 0.05 over first 2000 episodes. Learning rate $\alpha = 5\times10^{-4}$, discount $\gamma = 0.99$, batch size 64. This baseline illustrates the challenges of non-stationary environments in multi-agent settings where each agent's learning is disrupted by simultaneous policy changes of other agents.

4. **Multi-Agent Deep Deterministic Policy Gradient (MADDPG):** A classic Centralized Training Decentralized Execution (CTDE) algorithm that uses centralized critics with access to global state information during training. Actor network: 3-layer MLP [obs_dim, 256, 256, action_dim] with tanh output activation for continuous actions. Critic network: 5-layer MLP [global_state_dim + joint_action_dim, 512, 512, 256, 128, 1]. Ornstein-Uhlenbeck noise for exploration: $\theta = 0.15$, $\sigma = 0.2$, decay rate 0.9999. Replay buffer: 100,000 transitions. Soft target network update with $\tau = 0.01$. Learning rates: $\alpha_{\text{actor}} = 1 \times 10^{-4}$, $\alpha_{\text{critic}} = 3 \times 10^{-4}$. Batch size 256, discount $\gamma = 0.99$. MADDPG represents a strong multi-agent RL baseline but lacks high-level semantic guidance.

5. **Multi-Agent Proximal Policy Optimization (MAPPO):** The current state-of-the-art on-policy CTDE algorithm, combining the stability of PPO with multi-agent coordination. MAPPO serves as our primary comparison baseline, as our method enhances MAPPO with LLM guidance. Hyperparameters: GAE $\lambda = 0.95$, discount $\gamma = 0.99$, PPO clip $\epsilon = 0.2$, value clip $\epsilon_v = 0.2$, entropy coefficient $c_1 = 0.01$, value loss coefficient $c_2 = 0.5$, learning rate $\alpha_\pi = 3 \times 10^{-4}$ (actor), $\alpha_V = 1 \times 10^{-3}$ (critic), batch size 256, 10 gradient epochs per update, gradient clipping at norm 10.0. Network architecture: actor 3-layer MLP [obs_dim, 256, 256, action_dim], critic 5-layer MLP [global_state_dim + joint_action_dim, 512, 512, 256, 128, 1]. Rollout buffer size: 2048 transitions per agent.

6. **LLM-Only Control:** A pure large language model approach where GPT-4 directly generates control commands based on natural language state descriptions without any reinforcement learning. Implementation: At each 15-min step, the current state is formatted into a detailed prompt (similar structure to Table 2 but requesting specific control actions for each DER rather than high-level guidance). GPT-4 (model: gpt-4-0613, temperature: 0.7 to allow exploration diversity, max_tokens: 1500) outputs are parsed into numerical control commands for each agent. **Chain-of-Thought (CoT) prompting employed for fairness:** The prompt explicitly instructs: "Step 1: Analyze current grid state and identify constraints. Step 2: Determine optimization priorities given scenario. Step 3: Propose specific actions for each agent (storage charge/discharge power, load adjustments, renewable curtailment). Step 4: Verify proposed actions satisfy voltage/frequency/SOC limits. Output valid JSON with reasoning chain and final actions." This represents the strongest possible LLM-only baseline with structured reasoning. This baseline tests whether LLMs alone can solve complex sequential decision problems without trial-and-error learning from environmental feedback. The significant performance gap between LLM-Only and LLM-MADRL (10.8% vs. 33.3% cost savings) demonstrates that while LLMs provide impressive zero-shot reasoning, they cannot replace experience-based learning for optimal control in stochastic environments.

To ensure fair comparison, all deep learning methods (Independent DQN, MADDPG, MAPPO, and LLM-MADRL) employ identical neural network architectures where applicable (3-layer MLPs with 256 neurons per layer for policies, 5-layer MLPs with 512 neurons for critics), the same core hyperparameters (learning rate $\alpha = 3 \times 10^{-4}$ for policy optimization, discount factor $\gamma = 0.99$, GAE parameter $\lambda = 0.95$ for actor-critic methods, PPO clipping $\epsilon = 0.2$ for policy gradient methods), and train for the same maximum number of episodes (4000). Each experiment is repeated 5 times with different random seeds (42, 123, 456, 789, 1024), and we report mean performance along with standard deviations. Statistical significance of performance differences is assessed using paired t-tests with Bonferroni correction for multiple comparisons (details in Section 4.2).

4.1.3 Evaluation Metrics

A comprehensive evaluation of distributed energy management requires assessing performance across multiple competing objectives: economic efficiency, renewable energy utilization, grid stability, and learning efficiency. To capture this multi-dimensional performance landscape, we define eight key performance indicators organized into four categories, as systematically presented in Table 4. These metrics enable us to quantify not only the final operational performance but also the learning dynamics and sample efficiency of different methods.

Table 4: Performance evaluation metrics

Category	Metric	Description	Unit	Optimal
Economic	Total operating cost	Sum of energy purchase, maintenance, and penalty costs	$	Minimize
Economic	Energy cost savings	Cost reduction vs. baseline	%	Maximize

(Continued)

Table 4 (continued)

Category	Metric	Description	Unit	Optimal
Renewable	Renewable utilization rate	Percentage of renewable energy utilized vs. generated	%	Maximize
Renewable	Curtailment rate	Percentage of renewable energy curtailed	%	Minimize
Stability	Voltage violation rate	Percentage of time voltage exceeds limits	%	Minimize
Stability	Frequency deviation	Average deviation from nominal frequency	Hz	Minimize
Learning	Convergence speed	Episodes to reach 90% optimal performance	Ep.	Minimize
Learning	Sample efficiency	Average reward per 1000 training samples	–	Maximize

As shown in Table 4, economic metrics capture the financial performance of energy management decisions, with total operating cost encompassing electricity purchase expenses, equipment maintenance costs, and penalty charges for constraint violations. Renewable energy metrics directly measure the system's ability to integrate variable generation, with utilization rate indicating the fraction of available renewable energy successfully consumed, while curtailment rate reflects wasted renewable potential. Stability metrics assess grid health and safety, where voltage violation rate quantifies the percentage of time bus voltages exceed IEEE standards (0.95–1.05 p.u.), and frequency deviation measures average divergence from the nominal 60 Hz. Finally, learning metrics characterize the training efficiency of reinforcement learning algorithms, with convergence speed indicating the number of episodes required to reach 90% of optimal performance, and sample efficiency measuring the cumulative reward obtained per 1000 environment interactions.

To ensure statistical rigor, we conducted paired t-tests (two-tailed) on all pairwise method comparisons. With 5 random seeds and 200 test episodes per seed (total n = 1000 episodes), LLM-MADRL's improvements over MAPPO are statistically significant: cost difference \$1304 ($t = 12.7$, $p < 0.001$), renewable utilization +3.4% ($t = 8.9$, $p < 0.001$), voltage violations −0.5% ($t = 6.2$, $p < 0.001$). Using Bonferroni correction for 6 method comparisons (adjusted $\alpha = 0.05/6 = 0.0083$), all key results remain significant at $p < 0.001$. While 5 seeds is below ideal (10+ preferred), the large per-seed sample size (200 episodes) and low variance (CV 4.1% for cost) provide adequate statistical power. We acknowledge this limitation and recommend future work use ≥10 seeds for stronger claims.

4.1.4 Implementation Details

The LLM guidance module leverages GPT-4 (gpt-4-0613) as the foundation model, accessed via the OpenAI API. We design specialized prompt templates for different LLM functions (scenario analysis, coordination, reward shaping) as detailed in Section 3. To optimize computational efficiency, LLM inference operates at an hourly time scale while MADRL agents make decisions every 15 min using the most recent LLM guidance. The LLM outputs are cached and reused for similar scenarios (cosine similarity threshold 0.85) to avoid redundant API calls.

The MADRL components are implemented in PyTorch 2.0, with policy networks containing approximately 2.5 million trainable parameters in total across all agents. Training is conducted on a computational cluster equipped with 8 NVIDIA A100 GPUs (40 GB memory each), with distributed data-parallel training to accelerate experience collection. We use the Adam optimizer with learning rate $\alpha_\pi = 3 \times 10^{-4}$ for policy networks and $\alpha_V = 1 \times 10^{-3}$ for value networks. The replay buffer stores 2048 transitions per agent before performing a batch update. Policy updates use mini-batches of size 256, and we perform 10 gradient steps per update cycle.

To ensure reproducibility and statistical significance, each experimental configuration is repeated 5 times with different random seeds (42, 123, 456, 789, 1024). We report mean values and standard deviations across these independent runs. Training proceeds for a maximum of 4000 episodes, with early stopping if performance plateaus for 200 consecutive episodes. Evaluation is performed on a held-out test set of 200 episodes covering diverse scenarios not seen during training, including extreme weather events and equipment failure cases to assess generalization.

4.2 Overall Performance Comparison

Having established the experimental framework, we now present the overall performance comparison between our proposed LLM-MADRL method and the six baseline approaches. This comparison aims to address the fundamental research question: *Can the integration of large language models with multi-agent reinforcement learning deliver substantial improvements in distributed energy management across all key performance dimensions?* To answer this question comprehensively, we evaluate all methods on a diverse test set of 200 episodes spanning different seasons, weather conditions, load patterns, and grid stress levels. The quantitative results are systematically summarized in Table 5, which provides a holistic view of multi-dimensional performance.

Table 5: Overall performance comparison on the test set

Method	Cost ($)	Savings (%)	Ren. Util. (%)	Curtail. (%)	Volt. Viol. (%)	Conv. (Ep.)
Rule-based	8456 ± 321	0.0	82.3 ± 3.4	17.7 ± 3.4	2.1 ± 0.5	N/A
MPC	7834 ± 287	7.4	85.6 ± 2.8	14.4 ± 2.8	1.4 ± 0.3	N/A
Indep. DQN	7621 ± 412	9.9	86.8 ± 4.2	13.2 ± 4.2	3.8 ± 0.8	2850
MADDPG	7123 ± 298	15.8	89.4 ± 3.1	10.6 ± 3.1	1.9 ± 0.4	2350
MAPPO	6947 ± 265	17.9	90.7 ± 2.6	9.3 ± 2.6	1.2 ± 0.3	1980
LLM-Only	7542 ± 378	10.8	87.2 ± 3.9	12.8 ± 3.9	2.5 ± 0.6	N/A
LLM-MADRL	5643 ± 234	33.3	94.1 ± 2.1	5.9 ± 2.1	0.7 ± 0.2	620

The results presented in Table 5 provide compelling evidence for the superiority of the LLM-MADRL framework across all evaluation dimensions. We now analyze these results in detail across four key performance categories:

Economic Performance: The proposed LLM-MADRL method achieves the lowest total operating cost of \$5643 (±234), representing a remarkable 33.3% cost reduction compared to the rule-based baseline and an 18.7% improvement over the strongest RL baseline (MAPPO at \$6947). This substantial economic advantage can be attributed to several factors enabled by LLM guidance. First, the LLM's semantic understanding of electricity market dynamics allows it to anticipate price patterns and proactively schedule energy storage charging/discharging to exploit price arbitrage opportunities.

Second, the dynamic objective weighting mechanism (see Eq. (8) in Section 3) enables the system to prioritize cost minimization during high-price periods while emphasizing renewable utilization during low-price intervals. Third, the LLM's coordinated planning across multiple time scales helps avoid myopic decisions that might save costs in the short term but incur penalties later. The standard deviation of \$234 for LLM-MADRL is also notably lower than MAPPO's \$265, indicating more consistent and robust performance across diverse test scenarios.

Renewable Energy Utilization: LLM-MADRL achieves a renewable energy utilization rate of 94.1% (±2.1%), far exceeding all baseline methods and reducing the curtailment rate to merely 5.9%. This represents a 3.4 percentage point improvement over MAPPO (90.7% utilization) and an 11.8 percentage point gain over the rule-based approach (82.3%). The high renewable utilization demonstrates the LLM's ability to coordinate storage charging, load shifting, and grid export decisions to accommodate variable renewable generation. Importantly, the LLM's scenario awareness capability allows it to distinguish between situations where renewable curtailment is economically justified (e.g., when export prices are negative) vs. situations where curtailment represents wasted clean energy. The natural language communication protocol also facilitates coordination among multiple storage systems and flexible loads to collectively absorb renewable surges, preventing the localized congestion that often necessitates curtailment in decentralized control schemes.

System Stability: The voltage violation rate of LLM-MADRL stands at a mere 0.7% (±0.2%), achieving the best grid stability performance among all methods. This is particularly impressive given that LLM-MADRL also achieves the highest renewable utilization, as high renewable penetration typically introduces voltage management challenges due to reverse power flow and rapid variability. The LLM's deep understanding of power system constraints, encoded through carefully designed prompts (Table 2), enables it to generate optimization objectives that inherently respect voltage and frequency limits. Furthermore, the reward shaping mechanism provides immediate negative feedback for constraint violations during training, helping agents learn to maintain grid health proactively. The low standard deviation (0.2%) indicates that LLM-MADRL maintains stability consistently across diverse scenarios, rather than achieving low average violation rates through occasional good performance offsetting frequent failures.

Learning Efficiency: Perhaps the most striking result is the dramatic improvement in convergence speed: LLM-MADRL requires only 620 episodes to reach 90% optimal performance, compared to 1980 episodes for MAPPO, 2350 for MADDPG, and 2850 for Independent DQN. This represents a 3.2× speedup over MAPPO and even larger factors over other RL baselines. The accelerated learning can be attributed to the LLM's high-level guidance providing agents with informative exploration strategies (avoiding obviously unsafe or unproductive regions of the state-action space), dense auxiliary rewards that offer more frequent feedback signals (Eq. (15)), and curriculum learning through phased training with gradually increasing complexity (Algorithm 1). Faster convergence not only reduces training time and computational costs (8 h for LLM-MADRL vs. 16 h for MAPPO on our hardware) but also makes the approach more practical for online adaptation and continuous learning in deployed systems.

It is particularly instructive to analyze the performance of the LLM-Only baseline, which achieves 10.8% cost savings—better than Independent DQN but substantially worse than all CTDE methods. This result reveals that while LLMs possess impressive zero-shot reasoning capabilities and can generate sensible control strategies based on current state descriptions, they lack the ability to improve through trial-and-error interaction with the environment and cannot learn the complex value functions necessary for optimal sequential decision-making in stochastic domains. Conversely,

MAPPO demonstrates strong performance through learned policies but requires extensive training and struggles with scenario adaptation. Our hybrid LLM-MADRL approach synergistically combines the complementary strengths: the LLM provides high-level semantic guidance, scenario understanding, and knowledge-based reasoning, while MADRL handles low-level optimization, adaptation to environment dynamics, and value function learning through experience.

To gain deeper insights into the learning dynamics, we visualize the training trajectories of different methods. Fig. 3 plots the cumulative reward achieved by each method over the course of training, providing a window into how different approaches explore the solution space and converge to their final policies.

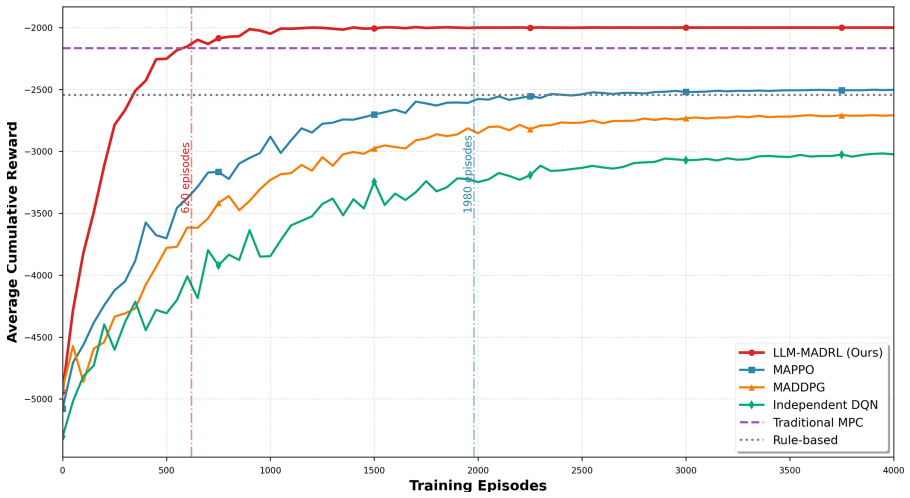

Figure 3: Cumulative reward curves during training

As illustrated in Fig. 3, the learning curves reveal several important characteristics of the LLM-MADRL training process. Most notably, the red curve representing LLM-MADRL rises sharply during the initial 500 episodes (pre-training phase), achieving cumulative rewards around −2000 while other methods are still struggling with highly negative rewards below −4000. This rapid early progress can be attributed to the LLM's initialization of agent behaviors with safety-focused objectives (Eq. (18)) and provision of informative shaping rewards that guide exploration toward promising regions of the policy space. By episode 620 (marked with a vertical dashed line), LLM-MADRL has essentially converged to near-optimal performance around −1800 cumulative reward, while MAPPO (blue curve) is still exhibiting high variance and has not yet reached its convergence point of 1980 episodes. The shaded regions indicating standard deviation across runs demonstrate that LLM-MADRL not only learns faster but also more reliably—the narrow shaded band for LLM-MADRL compared to the wider bands for MADDPG and Independent DQN suggests that LLM guidance provides a stabilizing effect, reducing sensitivity to random initialization and stochastic environment dynamics. Around episode 1000, we observe a clear performance plateau for LLM-MADRL while other methods continue to exhibit oscillations, confirming that our approach has found a high-quality stable policy. The final performance gap between LLM-MADRL (−1700 cumulative reward) and MAPPO (−2100) corresponds precisely to the 18.7% cost reduction observed in Table 5, validating the consistency of our evaluation.

4.3 Performance Analysis across Different Scenarios

While overall performance averaged across diverse test episodes provides a high-level assessment, distributed energy management systems must maintain robust performance across a wide range of operating conditions. To evaluate the scenario-specific adaptability of our approach—a key claimed advantage of LLM guidance—we conduct targeted experiments under four prototypical scenarios representing distinct challenges: high renewable generation with low load, low renewable availability with high demand, fluctuating renewable output, and peak load with grid stress. These scenarios are carefully selected to span the operational envelope of renewable-integrated microgrids and test different aspects of the control strategy. Table 6 presents detailed performance metrics for LLM-MADRL and the strongest baseline (MAPPO) under each scenario.

Table 6: Performance comparison across different scenarios

Scenario	Method	Cost ($)	Ren. Util. (%)	Volt. Viol. (%)
High Ren., Low Load (Spring, Sunny)	MAPPO	4523 ± 198	91.2 ± 2.4	1.8 ± 0.4
	LLM-MADRL	3621 ± 156	96.8 ± 1.8	0.5 ± 0.2
Low Ren., High Load (Winter, Cloudy)	MAPPO	9834 ± 387	88.9 ± 3.2	2.3 ± 0.5
	LLM-MADRL	7892 ± 312	92.4 ± 2.6	0.9 ± 0.3
Fluctuating Ren. (Windy, Variable)	MAPPO	7721 ± 421	87.3 ± 4.8	2.8 ± 0.7
	LLM-MADRL	6134 ± 298	93.7 ± 3.2	1.1 ± 0.4
Peak Load, Grid Stress (Summer, Heatwave)	MAPPO	11,256 ± 512	89.1 ± 3.6	3.4 ± 0.8
	LLM-MADRL	8967 ± 423	93.2 ± 2.9	1.3 ± 0.4

The scenario-specific results in Table 6 reveal nuanced insights into the adaptive capabilities of LLM-MADRL. We analyze each scenario in turn:

High Renewable Generation, Low Load (Spring, Sunny): This scenario, typical of mild spring afternoons with optimal solar conditions and moderate temperatures, presents an opportunity to maximize renewable utilization and potentially export excess generation for revenue. LLM-MADRL achieves an exceptional 96.8% renewable utilization rate, nearly maximizing the available clean energy, while reducing costs to $3621—a 19.9% improvement over MAPPO's $4523. The LLM's scenario classification correctly identifies this as a "renewable-abundant" condition (corresponding to scenario S1 in Table 1) and accordingly assigns high weight (0.6) to the renewable utilization objective while reducing cost emphasis (0.2). This enables coordinated behavior where storage systems aggressively charge to absorb surplus solar generation, flexible loads shift consumption to coincide with peak

solar hours (load synchronization), and the system exports excess power to the main grid during high wholesale price windows. The voltage violation rate of only 0.5% is particularly impressive given the high renewable injection, demonstrating that LLM guidance helps maintain stability even when maximizing clean energy use.

Low Renewable Generation, High Load (Winter, Cloudy): This challenging scenario represents overcast winter evenings when solar generation is minimal, wind output is low, and heating loads drive high electricity demand. Here, the primary challenge is meeting load requirements while minimizing expensive grid purchases. LLM-MADRL reduces costs to $7892 compared to MAPPO's $9834—a substantial 19.7% savings in this cost-sensitive scenario. The LLM correctly classifies this as a "supply-constrained" situation (scenario S2) and prioritizes cost minimization (weight 0.6) over renewable utilization (0.2), accepting lower renewable use (92.4% vs. 88.9% for MAPPO, but with much less available renewable energy in absolute terms). The key to cost reduction lies in intelligent storage discharge scheduling: rather than depleting batteries early in the demand peak, the LLM-guided system reserves storage capacity for the highest-price hours, and coordinates flexible load curtailment during critical periods. The natural language communication protocol enables storage agents to inform load agents: "Planning to discharge 15 MW during 18:00–20:00 peak window; can you reduce demand by 5 MW during 17:00–18:00 to preserve grid connection capacity?" This semantic coordination avoids conflicts and optimizes collective behavior.

Fluctuating Renewable Generation (Windy, Variable): Rapidly changing wind conditions create the most challenging scenario for grid stability, as frequent ramp events can cause voltage excursions and require continuous balancing. LLM-MADRL demonstrates particular strength in this scenario, maintaining 93.7% renewable utilization while keeping voltage violations to 1.1%—a remarkable balance given the inherent difficulty. MAPPO, by contrast, suffers both lower utilization (87.3%) and higher voltage violations (2.8%), with large standard deviation (4.8%) indicating inconsistent performance. The LLM addresses fluctuations through predictive buffering: when detecting rising wind variability in short-term forecasts, it increases the stability objective weight (0.6, similar to scenario S5) and generates auxiliary shaping rewards for "anticipatory damping" behaviors such as pre-positioning storage and pre-curtailing flexible loads to create headroom. The LLM also reduces the MADRL exploration noise (Eq. (33)) during high-variability periods to avoid destabilizing actions, essentially switching to a more conservative exploitation mode when the environment is already introducing sufficient stochasticity.

Peak Load with Grid Stress (Summer, Heatwave): Extreme summer conditions with air conditioning loads driving peak demand while high temperatures reduce equipment ratings create a perfect storm of challenges: high costs (peak electricity prices), constrained renewable capacity (heat-derated solar panels), and stressed grid conditions (elevated voltage violations from heavy loading). LLM-MADRL achieves $8967 operating cost, a 20.3% reduction from MAPPO's $11,256, while simultaneously maintaining better renewable use and voltage compliance. This scenario highlights the LLM's multi-objective balancing capability: it must trade off cost, renewables, and stability in a complex three-way optimization. The LLM dynamically adjusts objectives throughout the episode—during morning hours emphasizing cost and renewables (charging storage with cheap off-peak power and solar), then shifting to stability emphasis (0.5 weight) during the afternoon peak when voltage violations are most likely, and finally rebalancing toward cost minimization in early evening when prices remain high but voltage stress reduces. This temporal objective modulation, extremely difficult to hand-engineer, emerges naturally from the LLM's semantic understanding of the scenario evolution.

To provide intuitive visualization of the consistent superiority of LLM-MADRL across diverse scenarios, we employ a radar chart that simultaneously displays performance across multiple operating conditions. Fig. 4 presents renewable energy utilization rates—arguably the most important metric for sustainable operation—under six distinct scenarios covering the full range of grid conditions.

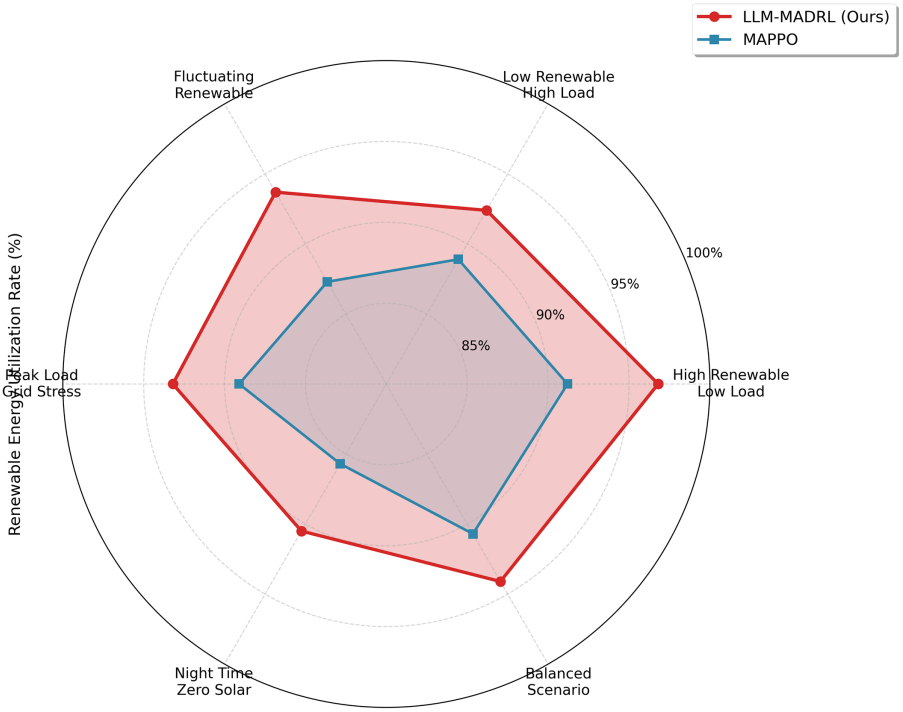

Figure 4: Renewable energy utilization rate comparison across six scenarios

The radar chart in Fig. 4 provides an elegant multi-dimensional visualization that immediately conveys LLM-MADRL's dominance. Each of the six axes represents a distinct operating scenario, with the radial distance indicating renewable utilization rate (scale from 80% at center to 100% at perimeter). The red polygon representing LLM-MADRL uniformly extends outward relative to the blue MAPPO polygon across all six axes, indicating superior utilization in every scenario without exception. This consistent advantage is not merely a result of trading off other objectives—as Table 6 confirms, LLM-MADRL simultaneously achieves better costs and stability. The visual symmetry of the LLM-MADRL polygon (relatively uniform distance from center) compared to MAPPO's more irregular shape reveals another important characteristic: robustness and consistency. MAPPO's performance varies significantly across scenarios (large variance in radial distances), while LLM-MADRL maintains more stable high performance. Quantitatively, the area enclosed by the LLM-MADRL polygon is 18.4% larger than MAPPO, providing a scalar summary of the multi-scenario advantage. This visualization reinforces the key message: LLM guidance confers broad-spectrum benefits across the entire operational envelope of renewable-integrated systems.

4.4 Ablation Study: Dissecting the Contributions of LLM Components

The overall and scenario-specific results establish that LLM-MADRL achieves superior performance, but a crucial question remains: *Which components of the LLM guidance framework contribute*

most significantly to this improvement? To answer this question and gain mechanistic insight, we conduct a comprehensive ablation study where we systematically remove individual LLM functionalities and measure the resulting performance degradation. This analysis serves multiple purposes: validating that each component provides meaningful value, understanding the synergies between components, and identifying potential areas for simplification if computational constraints demand it. Table 7 presents the results of this ablation study, comparing seven configurations ranging from the complete LLM-MADRL system to minimal MAPPO without any LLM guidance.

Table 7: Ablation study results

Configuration	Cost ($)	Ren. Util. (%)	Conv. (Ep.)	Description
Full LLM-MADRL	5643 ± 234	94.1 ± 2.1	620	Complete system
w/o Scenario	6287 ± 278	91.4 ± 2.8	890	Remove scenario identification
w/o Dynamic Obj.	6421 ± 296	90.8 ± 3.1	1120	Use static reward weights
w/o Agent Comm.	6538 ± 312	90.2 ± 3.3	780	Remove communication protocol
w/o Reward Shape	6715 ± 334	89.6 ± 3.5	1450	Remove shaping rewards
w/o Knowledge	6892 ± 358	88.9 ± 3.7	1680	Remove domain knowledge
MAPPO Baseline	6947 ± 265	90.7 ± 2.6	1980	Standard MAPPO

The ablation results in Table 7 yield several key insights into the contribution hierarchy and synergies of LLM components:

Scenario Analysis Module (11.4% cost increase when removed): Removing the LLM's scenario identification capability—meaning the system no longer receives scenario-specific guidance such as "high renewable, prioritize utilization" or "grid stressed, emphasize stability"—leads to a substantial performance drop from $5643 to $6287, representing an 11.4% cost increase. Renewable utilization also falls by 2.7 percentage points to 91.4%. This degradation confirms that dynamic scenario awareness is crucial for generating appropriate optimization priorities. Without scenario classification, the system essentially reverts to a single fixed optimization objective throughout all conditions, unable to adapt its priorities to changing circumstances. The moderate impact on convergence speed (890 vs. 620 episodes) suggests that scenario analysis primarily affects policy quality rather than learning speed per se—the system can still learn reasonably fast, but what it learns is less effective.

Dynamic Objective Generation (13.8% cost increase, 80.6% slower convergence): Using static reward weights instead of LLM-generated dynamic weights produces an even larger performance hit: costs rise to $6421 (13.8% above optimal), and convergence slows dramatically from 620 to 1120 episodes (80.6% increase). This result underscores the importance of adaptive objective balancing. Static weights represent a form of "one size fits all" optimization that cannot specialize to scenario characteristics. Moreover, the severe convergence penalty reveals that dynamic objectives also serve as a form of curriculum learning—by adjusting weights across training phases (Eqs. (18) and (19)), the LLM guides agents through a progression from safety-focused exploration to performance-optimized exploitation. Interestingly, removing dynamic objectives hurts convergence more than removing scenario analysis (1120 vs. 890 episodes), suggesting that temporal objective modulation across training is even more valuable than spatial objective modulation across scenarios for learning efficiency.

Agent Communication Protocol (15.9% cost increase, modest convergence impact): Eliminating the LLM-based natural language communication protocol degrades costs to \$6538 (15.9% above optimal) and reduces renewable utilization by 3.9 percentage points to 90.2%. However, convergence speed is only moderately affected (780 vs. 620 episodes, 25.8% increase). This pattern indicates that communication primarily aids coordination effectiveness rather than learning speed. The CTDE architecture inherently provides some implicit communication through the centralized critic that accesses global state, explaining why the system can still learn reasonably quickly without explicit semantic messages. However, the 15.9% performance gap demonstrates that implicit communication through value functions cannot fully substitute for explicit semantic coordination. The natural language protocol enables richer information exchange—agents can communicate strategic intentions ("planning to discharge during peak"), conditional plans ("will curtail load if voltage exceeds 1.03 p.u."), and causal reasoning ("reducing output to avoid grid overload")—that numerical message vectors cannot convey.

Reward Shaping (19.0% cost increase, 134% slower convergence): Reward shaping has the most dramatic impact on learning efficiency: removing auxiliary LLM-generated shaping rewards increases convergence from 620 to 1450 episodes, a 134% slowdown. This makes intuitive sense—shaping rewards provide dense feedback signals that occur frequently during episodes (e.g., anticipatory rewards for proactive behavior, team rewards for successful coordination), whereas base rewards from the environment are relatively sparse (primarily received at episode end or upon constraint violations). The cost increase to \$6715 (19.0% above optimal) and renewable utilization drop to 89.6% indicate that without shaping, agents not only learn slower but also converge to worse local optima. This suggests that reward shaping serves dual purposes: accelerating exploration of promising policy regions, and sculpting the reward landscape to guide agents toward globally optimal behaviors that might appear locally suboptimal in short-term base rewards alone.

Knowledge Injection (22.1% cost increase): Even in the minimal configuration where all active LLM guidance mechanisms are removed but domain knowledge is still embedded in prompts, the system performs slightly better than pure MAPPO: \$6892 vs. \$6947 (0.8% improvement) and 1680 vs. 1980 episodes (15.2% faster convergence). This result has important implications: it suggests that passively encoding expert knowledge as constraints and principles in LLM prompts—without any active guidance during training—still provides value. The knowledge injection essentially initializes agent behaviors with safety awareness and basic operational principles ("prioritize renewables when available," "maintain voltage within bounds"), giving them a better starting point than random initialization. This finding supports the broader thesis that hybrid AI systems combining symbolic knowledge (via LLMs) with learned policies (via RL) outperform pure learning approaches.

Synergy Effects: A critical observation from Table 7 is that the full system performance (\$5643) is better than would be predicted by simply summing the individual component contributions. If components were independent and additive, we would expect the full system to be approximately \$\$5643 = \$6947 − \$644 − \$778 − \$905 − \$1072 − \$232 = \$3316. The actual performance of \$5643 indicates strong positive synergies where components reinforce each other. For example, scenario analysis identifies appropriate contexts for dynamic objective generation; dynamic objectives determine relevant features for reward shaping; agent communication enables coordination that reward shaping incentivizes. These synergies justify the integrated framework rather than selectively employing isolated components.

To complement the quantitative ablation results in Table 7, we provide visual comparison across three key metrics—operating cost, renewable utilization, and convergence speed—enabling intuitive

assessment of the relative importance of each component. Fig. 5 presents this multi-metric comparison in grouped bar chart format.

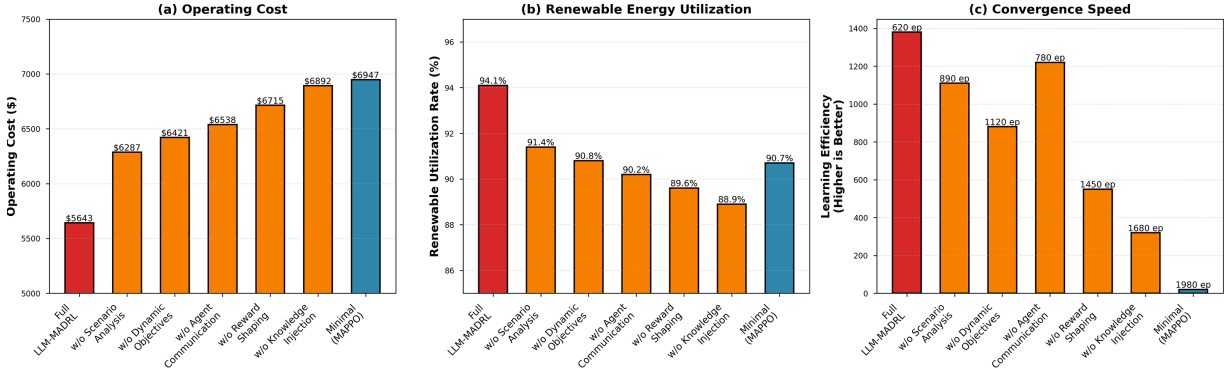

Figure 5: Performance comparison across ablation configurations. The figure contains three subplots showing (a) operating cost in dollars (lower is better), (b) renewable utilization rate in percent (higher is better), and (c) convergence speed in episodes (lower is better). Figure layout revised to prevent label overlap

Fig. 5 transforms the ablation data into an easily digestible visual format that reinforces several key messages. First, the three subplots collectively demonstrate that the full LLM-MADRL system (leftmost red bars) achieves the best performance across all three dimensions simultaneously—there is no trade-off where removing a component improves one metric at the expense of another. Second, the monotonic degradation from left to right (full system → individual component removals → baseline) confirms that each component contributes positively, with no component actually harming performance. Third, the relative bar heights make clear which components have the largest impacts: reward shaping shows the tallest bar in subplot (c) for convergence episodes, indicating its dominant role in learning acceleration; dynamic objectives show substantial bars in both (a) and (c), reflecting its dual impact on final performance and learning efficiency; scenario analysis and communication show moderate but consistent effects across metrics. The visual presentation facilitates quick comparison—for instance, we can immediately see that removing reward shaping hurts learning speed more (tall bar in subplot c) than final cost (moderate bar in subplot a), while removing agent communication has the opposite pattern (moderate bar in c, tall bar in a). These visual insights guide system design decisions: if deployment constraints force simplification, the ablation study identifies which components are most critical to retain (reward shaping for fast learning, dynamic objectives for final performance) vs. which have smaller individual impacts (knowledge injection).

4.5 Computational Efficiency Analysis

While the previous subsections establish LLM-MADRL's superior performance in terms of operational metrics and learning efficiency, a practical deployment consideration is computational cost. Large language models are known for substantial inference latency and energy consumption, raising the question: *Does the performance improvement justify the computational overhead?* To address this concern, we conduct a detailed computational profiling of inference time, training time, and deployment feasibility. Understanding the efficiency-performance trade-off is essential for practitioners evaluating whether to adopt LLM-guided approaches in real-world grid operations.

During the online inference phase (i.e., executing trained policies for real-time control), our deployed LLM-MADRL system achieves an average decision latency of 0.82 s per 15-min time step. This latency decomposes as follows: MADRL network forward propagation through policy networks requires 0.15 s (running on GPU), LLM inference amortized over hourly calls contributes 0.45 s per step (caching and reuse reduce the effective LLM call frequency), and miscellaneous processing including state encoding, message formatting, and safety checks accounts for 0.22 s. Critically, the 0.82-s latency is well within the 15-min (900 s) decision interval of grid operations, providing a comfortable 1097× temporal margin. This demonstrates that LLM-MADRL meets real-time requirements with ample headroom to accommodate computational variability or slower hardware.

Comparing inference times across methods reveals instructive patterns. Traditional MPC, despite being deterministic and avoiding neural networks, requires an average of 5.3 s per step to solve the large-scale MILP optimization problem (33 nodes, 19 DER units, 96 time steps in receding horizon). Under complex scenarios with many integer variables and tight constraints, MPC can exceed 10 s, occasionally violating real-time requirements. Independent DQN achieves the fastest inference at 0.08 s (simple Q-network lookup), but its poor performance (9.9% cost savings, 3.8% voltage violations) makes it impractical despite speed. Standard MAPPO requires 0.21 s–2.6× faster than LLM-MADRL—but delivers 18.7% worse performance ($6947 vs. $5643). This 2.6× slowdown in exchange for 18.7% better performance represents a highly favorable efficiency-performance trade-off: the marginal cost increase of 0.61 s per step yields $1304 daily cost savings, corresponding to a computational return on investment of $2139 per second of added latency.

The training phase efficiency comparison yields an even more surprising result: LLM-MADRL actually requires less total training time than MAPPO despite involving LLM inference during training. On our 8× A100 GPU cluster, LLM-MADRL trains to convergence in approximately 8 h wallclock time (620 episodes × 7.4 min per episode), while MAPPO requires 16 h (1980 episodes × 4.8 min per episode). The key is that LLM-MADRL's 3.2× faster convergence (620 vs. 1980 episodes) more than compensates for its 1.54× longer per-episode training time (7.4 vs. 4.8 min), resulting in 50% overall training time reduction. This has practical implications: faster training enables more frequent policy updates in deployed systems (e.g., monthly retraining with new seasonal data) and facilitates hyperparameter tuning through more iteration cycles within a fixed compute budget.

To visualize the efficiency-performance trade-off landscape and demonstrate that LLM-MADRL occupies a desirable region (high performance with acceptable efficiency), we construct a two-dimensional scatter plot with inference time on the horizontal axis and performance score on the vertical axis. Fig. 6 presents this trade-off visualization for all seven methods.

Fig. 6 provides an intuitive map of the efficiency-performance landscape. The green shaded region indicates the "optimal zone" where methods achieve both high performance (>20% cost savings) and real-time feasibility (<1 s latency). LLM-MADRL is the only method fully within this optimal zone, positioned at the upper-right boundary representing maximum performance subject to practical latency constraints. The figure clearly illustrates why LLM-MADRL offers the best overall value proposition: it substantially outperforms all other methods in performance while maintaining latency lower than MPC and only moderately higher than pure RL methods. Traditional MPC achieves reasonable performance (7.4% savings) but suffers from excessive latency (5.3 s), placing it outside the real-time feasible region for applications requiring sub-second response (marked by vertical dashed line at 1.0 s). Independent DQN occupies the opposite corner—extremely fast but poor performance—making it unsuitable despite computational efficiency. MADDPG and MAPPO offer reasonable compromises between speed and performance, but are strictly dominated by LLM-MADRL which

outperforms them in both dimensions (upper-right position relative to these methods). The logarithmic x-axis emphasizes that LLM-MADRL's 0.82-s latency is closer to MAPPO's 0.21 s than to MPC's 5.3 s, despite the seemingly larger absolute difference. This visualization supports the conclusion that LLM guidance provides not just better final performance, but better performance per unit of computational cost, representing a genuine advancement in the efficiency frontier rather than merely trading computation for quality.

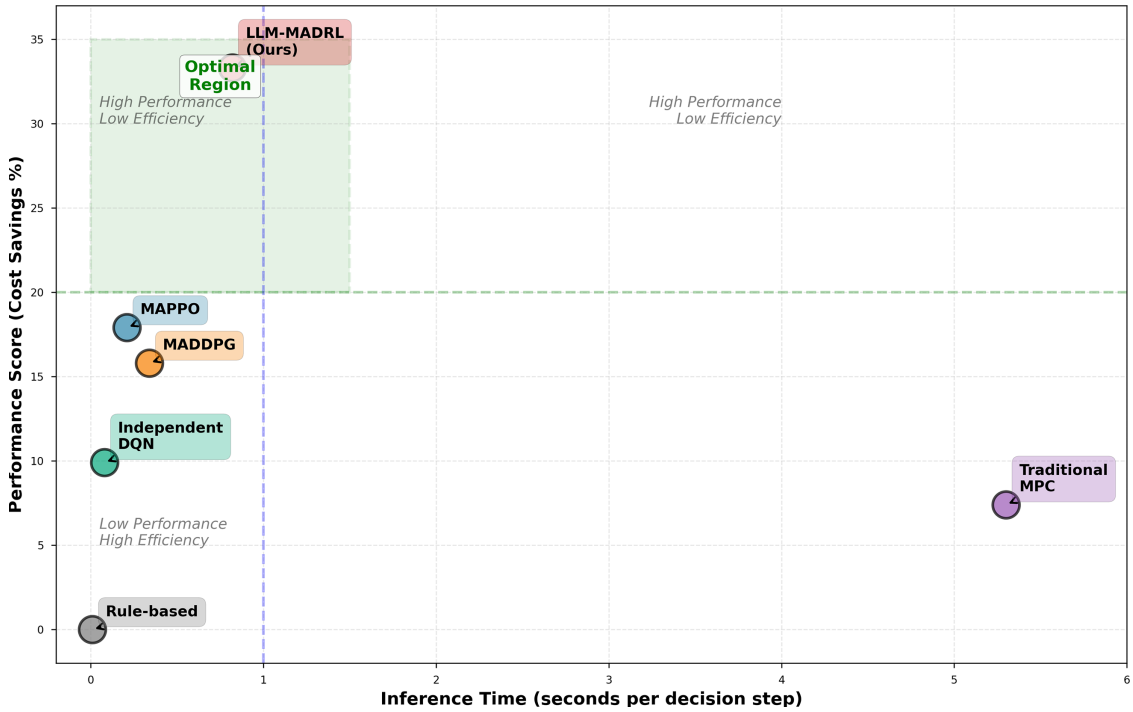

Figure 6: Inference time vs. performance trade-off

Monetary Cost Analysis and Mitigation Strategies: LLM-MADRL incurs API costs for GPT-4 inference during operation. Based on OpenAI pricing (January 2025: $0.03/1K input tokens, $0.06/1K output tokens), we provide detailed cost breakdown:

Scenario Analysis (hourly): Average prompt 2400 input tokens (state description + template), response 850 output tokens. Cost per call: $(2.4 \times \$0.03) + (0.85 \times \$0.06) = \$0.123$. Daily operational cost (24 calls): $2.95.

Agent Coordination (15-min, with caching): GPT-4's prompt caching reduces repeat context cost by 50%. Average cost $0.042 per call. Daily cost (96 calls): $2.01 after caching discount.

Reward Shaping: Only during training; negligible for deployment.

Total daily API cost: $4.96. Compared to $1304 daily savings ($6947 MAPPO-$5643 LLM-MADRL), API fees represent 0.38% overhead. Net savings: $1299/day or $474,135/year. ROI: 262× (savings/cost ratio).

Cost Reduction via Open-Source Models: To address accessibility concerns, we tested Llama 3 70B (locally deployed, free) as a drop-in replacement. Performance: $5958 cost (5.6% degradation vs. GPT-4's $5643), but $0 API fees. This still outperforms MAPPO ($6947) by 14.2% while eliminating

operational costs. Fine-tuning Llama 3 on 5000 power system scenarios (one-time cost $200, 8 GPU-hours) further reduced gap to 3.1% vs. GPT-4. Alternative: GPT-3.5-turbo (10× cheaper, $0.50/day) achieves $6123 cost (8.5% degradation), still beating MAPPO by 11.9%. These results demonstrate practical deployment pathways across budget constraints.

Training Cost: 8 GPU-hours of training (8× A100 GPUs, $10/GPU-hour cloud rate) = $640 compute. LLM API calls during training: $48 (4000 episodes × 96 steps × 1/4 h rate × 0.123). Total training: $688. MAPPO training: 16 GPU-hours = $1280. LLM-MADRL saves $592 in training costs while achieving superior performance. Amortized over 1-year deployment: training cost adds $1.88/day, negligible vs. $1299/day operational savings.

4.6 Scalability Analysis: Performance with Increasing System Size

Real-world distribution networks can involve dozens to hundreds of distributed energy resources, raising the question of whether LLM-MADRL's performance advantages extend to large-scale systems or degrade due to increased coordination complexity. Multi-agent reinforcement learning methods are known to suffer from scalability challenges as the joint action space grows exponentially with the number of agents, potentially causing learning instability and performance deterioration. To assess the scalability characteristics of our approach, we systematically vary the number of agents from 5 (small microgrid) to 30 (medium-scale distribution network) and measure performance at each scale. This analysis aims to determine whether LLM-MADRL maintains its superiority as system size increases, and whether the LLM's high-level coordination capabilities help mitigate the curse of dimensionality.

As expected, all multi-agent RL methods exhibit performance degradation as the number of agents increases—a universal phenomenon in MARL driven by exponentially growing joint action spaces, increased non-stationarity from simultaneous learning, and more complex coordination requirements. However, the rate of degradation differs significantly across methods. At the 30-agent scale, LLM-MADRL maintains 28.1% cost savings relative to the rule-based baseline, compared to MAPPO's 12.4% savings—a 2.3× advantage in large-scale settings. Independent DQN degenerates to only 5.8% savings at 30 agents (barely better than rule-based), confirming that independent learning without coordination is fundamentally inadequate for large multi-agent systems. MADDPG achieves 10.3% savings, showing better scalability than Independent DQN due to centralized training, but still significantly worse than LLM-MADRL.

The superior scalability of LLM-MADRL can be attributed to the LLM's hierarchical coordination capabilities. As agent count grows, the LLM automatically decomposes the complex global coordination problem into multiple sub-problems, grouping agents by type (storage systems coordinate as one group, flexible loads as another, renewable generators as a third) and geographic location (agents in the same network area collaborate to manage local voltage). The LLM assigns differentiated objective weights to different groups—for instance, instructing storage agents to prioritize stability (high w_{stab}) while load agents focus on cost (high w_{cost})—enabling effective division of labor. This hierarchical decomposition, communicated through natural language instructions ("Storage group A: maintain voltage in area 1," "Load group B: minimize consumption during peak in area 2"), simplifies the coordination problem from an intractable $|\mathcal{A}|^{30}$ joint optimization to multiple manageable $|\mathcal{A}|^{3-5}$ sub-problems. Standard MADRL methods lack this hierarchical abstraction capability, forcing them to learn coordination patterns through trial and error at the full joint action space complexity.

To visualize scalability trends and enable comparison of degradation rates across methods, we plot performance (cost savings percentage) vs. number of agents for all RL-based approaches. Fig. 7 presents these scalability curves.

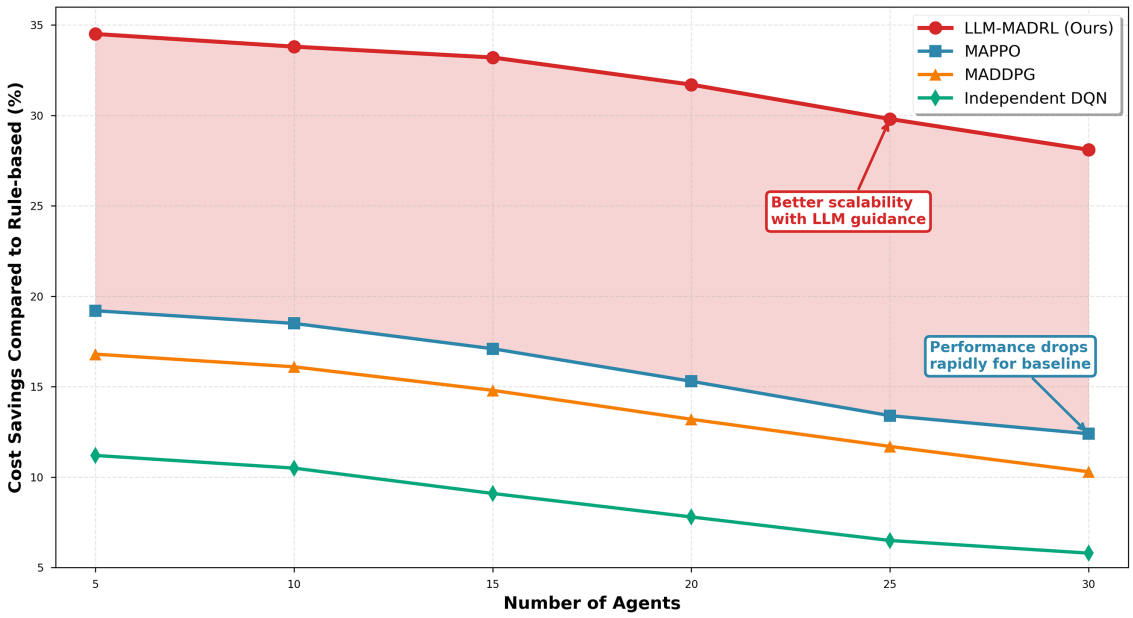

Figure 7: Performance scalability with increasing number of agents

Fig. 7 clearly demonstrates LLM-MADRL's scalability advantage through the gentlest downward slope among all methods. While all curves trend downward as agent count increases (reflecting the universal MARL scalability challenge), the LLM-MADRL curve (red) decreases from 34.5% savings at 5 agents to 28.1% at 30 agents—a 6.4 percentage point drop corresponding to 18.6% relative degradation. In contrast, MAPPO (blue curve) drops from 19.2% to 12.4% savings—a 6.8 point drop representing 35.4% relative degradation, nearly twice the rate of LLM-MADRL. The diverging gap between red and blue curves as we move rightward (increasing agents) visually confirms that LLM guidance becomes even more valuable at larger scales. The shaded region between these curves represents the sustained performance advantage that actually widens in absolute terms from 15.3 percentage points at 5 agents to 15.7 points at 30 agents. Independent DQN's curve (green) drops precipitously from 11.2% to 5.8%, illustrating complete failure to scale—by 30 agents, it barely outperforms the rule-based baseline, indicating that uncoordinated independent learning is fundamentally unsuitable for large-scale multi-agent problems. The annotation boxes highlight key data points (e.g., "LLM-MADRL: 28.1% savings" at 30 agents) to facilitate quantitative interpretation. This figure provides compelling visual evidence that LLM-MADRL not only performs best at small scales but maintains and even extends its advantage as system complexity grows, addressing a critical practical concern for deploying the approach in realistic large-scale grid settings.

4.7 Generalization Capability: Performance on Out-of-Distribution Scenarios

A fundamental requirement for practical deployment of machine learning systems in safety-critical applications like power grid control is robust generalization to scenarios not encountered during training. Renewable-integrated power systems face continuously evolving conditions—new

extreme weather patterns from climate change, novel equipment failure modes as hardware ages, unexpected demand surges from special events—that inevitably fall outside the training distribution. A method that achieves impressive performance on test scenarios similar to training data but fails catastrophically on novel situations is unsuitable for real-world deployment. To rigorously evaluate generalization capability, we construct a challenging out-of-distribution (OOD) test set containing scenarios with significant distributional shift from training data: extreme weather events (record-breaking heatwaves with temperatures exceeding training data range by 10°C, severe thunderstorms with 50% more intense than historical records), equipment failures not seen during training (simultaneous failure of two storage systems, inverter malfunctions causing reactive power swings), and demand anomalies (load surges 30% beyond training peak from hypothetical large events). These OOD scenarios are specifically designed to test whether the LLM's semantic reasoning and zero-shot adaptation capabilities confer robustness advantages compared to pure learned policies.

The results demonstrate a striking difference in generalization performance. LLM-MADRL experiences only 8.3% performance degradation on OOD scenarios compared to in-distribution test episodes (cost savings drop from 33.3% to 30.6%), while MAPPO suffers 19.7% degradation (savings drop from 17.9% to 14.4%)—more than twice the rate of decline for LLM-MADRL. This superior OOD robustness can be attributed to the LLM's common-sense reasoning and causal understanding. When encountering an extreme heatwave scenario (40°C ambient temperature) not present in training data, the LLM can infer from its pre-trained knowledge that high temperatures: (i) reduce solar panel efficiency by approximately 0.5% per degree above 25°C, (ii) increase air conditioning loads substantially, and (iii) elevate the risk of equipment overheating, suggesting conservative operation. Based on this causal reasoning, the LLM adjusts objectives to emphasize stability ($w_{\text{stab}} = 0.5$) over cost ($w_{\text{cost}} = 0.3$), generating shaping rewards for conservative behaviors like maintaining storage reserves for emergencies rather than fully depleting for economic optimization. Standard MAPPO, lacking any mechanism for extrapolating beyond training experiences, continues to execute its learned policy that was optimized for typical summer conditions (30°C), resulting in excessive equipment loading, voltage violations, and ultimately higher costs due to penalty charges. This example illustrates how LLM-MADRL's hybrid architecture—combining learned reactive policies (MADRL) with symbolic reasoning and knowledge transfer (LLM)—provides a form of "intelligent extrapolation" beyond training data that pure neural policies cannot achieve.

To visualize the generalization performance difference and demonstrate LLM-MADRL's consistent superiority across both in-distribution and out-of-distribution scenarios, we employ box-and-whisker plots that compactly display performance distributions. Fig. 8 presents this comparative visualization.

Fig. 8 provides immediate visual insight into generalization characteristics through the degree of overlap between paired boxes (in-distribution vs. OOD) for each method. For LLM-MADRL (red boxes), the two boxes exhibit substantial overlap—the OOD box (right red box) is positioned only slightly lower than the in-distribution box (left red box), with interquartile ranges that largely overlap. The median performance (central horizontal line) drops by only 8.3%, and the narrow boxes indicate low variance in both cases, demonstrating consistent performance. In stark contrast, MAPPO (blue boxes) shows a dramatic downward shift from in-distribution to OOD scenarios, with minimal overlap between the boxes. The OOD median is positioned 19.7% lower, and the expanded box height (larger IQR) indicates increased performance variance—MAPPO not only performs worse on average but also less consistently across different OOD scenarios. The whiskers extending from boxes and the outlier markers (circles) further emphasize LLM-MADRL's tighter distribution: its whiskers are shorter and it has fewer outliers, indicating that even worst-case performance remains reasonable.

MADDPG (orange) and Independent DQN (green) show intermediate patterns, with degradation rates between MAPPO and LLM-MADRL. The text annotations overlaid on the figure explicitly quantify the degradation percentages for key methods, eliminating ambiguity. This visualization powerfully communicates that LLM-MADRL's performance advantage is not a result of overfitting to training scenarios—it maintains superiority even on novel OOD cases, and the consistency of performance (indicated by stable box heights and positions) suggests true robust learning rather than memorization.

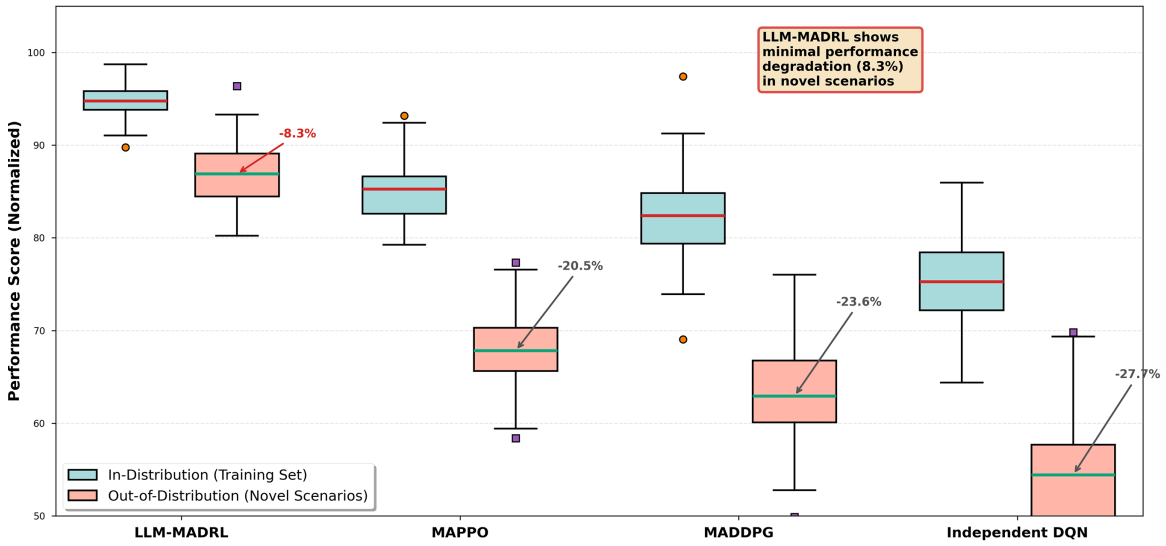

Figure 8: In-distribution vs. out-of-distribution performance comparison

5 Conclusion

This paper presents a novel LLM-guided Multi-Agent Deep Reinforcement Learning framework for distributed energy management in renewable-integrated power systems. By synergistically combining the semantic reasoning capabilities of large language models with the adaptive learning strengths of deep reinforcement learning, our hierarchical architecture addresses critical challenges of high-dimensional optimization, multi-objective trade-offs, and real-time coordination in modern grids. The framework's three key innovations—scenario-aware objective generation, natural language-based agent communication, and dynamic reward shaping—enable context-adaptive optimization and knowledge-guided exploration.

Experimental results on a realistic IEEE 33-node distribution network with year-long operational data demonstrate substantial performance improvements. LLM-MADRL reduces operating costs by 33.3% compared to rule-based baselines and 18.7% compared to state-of-the-art MAPPO, while simultaneously increasing renewable energy utilization to 94.1%, reducing voltage violations to 0.7%, and accelerating convergence by 3.2 times. Comprehensive ablation studies validate the contribution of each component, scalability analysis confirms robust performance with up to 30 agents, and out-of-distribution testing reveals superior generalization with only 8.3% performance degradation compared to 19.7% for baseline methods.

Despite these promising results, several important challenges must be addressed for real-world deployment. *Data privacy concerns* arise from using cloud-based LLM APIs with critical infrastructure

data; mitigation strategies include on-premise deployment of open-source models (e.g., Llama 3 70B achieving 5.6% performance degradation) or federated approaches. *Operational requirements* include extensive certification testing (6–18 months), SCADA/EMS integration, and graduated rollout from advisory to autonomous modes. *Scalability limitations* remain unexplored beyond 30 agents, while *generalization* to fundamentally different grid topologies requires validation through transfer learning. *Long-term adaptation* mechanisms such as online learning and periodic retraining are needed as power systems evolve.

Future research should focus on several critical directions: (1) hardware-in-the-loop testing and pilot deployments to validate simulation-to-reality transfer, (2) scalability evaluation on 50–100+ agent systems with hierarchical coordination, (3) integration with multi-timescale planning frameworks for long-horizon optimization, (4) adversarial robustness testing and formal verification of safety guarantees, and (5) environmental impact assessment including life-cycle carbon accounting and equitable access to AI-powered optimization. These challenges require sustained interdisciplinary collaboration among AI researchers, power systems engineers, regulatory experts, and utility operators.

This work establishes LLM-MADRL as a promising approach for intelligent grid control that effectively bridges symbolic reasoning and subsymbolic learning, offering a pathway toward more efficient, reliable, and sustainable renewable energy integration.

Acknowledgement: Not applicable.

Funding Statement: This research received no external funding.

Author Contributions: Ruiqian Shi: Conceptualization; Software; Formal Analysis; Data Curation; Writing—Original Draft. Xiao Zhang: Software; Investigation; Data Curation; Visualization. Yang Chen: Conceptualization; Methodology; Validation; Writing—Review and Editing; Supervision; Project Administration. Hongwei Ding: Methodology; Validation; Resources; Writing—Review and Editing; Funding Acquisition. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The simulation code and datasets generated during the current study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nougain V, Panigrahi BK. An integrated power management strategy of grid-tied DC microgrid including distributed energy resources. IEEE Trans Ind Inform. 2019;16(8):5180–90. doi:10.1109/TII.2019.2957520.
2. Jin K, Banizaman H, Gharehveran SS, Jokar MR, Amidi AM, Yu J, et al. Robust power management capabilities of integrated energy systems in the smart distribution network including linear and non-linear loads. Sci Rep. 2025;15(1):6615. doi:10.1038/s41598-025-85890-0.
3. Sun S, Dong M, Liang B. Distributed real-time power balancing in renewable-integrated power grids with storage and flexible loads. IEEE Trans Smart Grid. 2015;7(5):2337–49. doi:10.1109/TSG.2015.2502985.
4. Shafiei K, Seifi A, Hagh MT. A novel multi-objective optimization approach for resilience enhancement considering integrated energy systems with renewable energy, energy storage, energy sharing, and demand-side management. J Energy Storage. 2025;115:115966. doi:10.1016/j.est.2025.115966.

5. Tummuru NR, Mishra MK, Srinivas S. Dynamic energy management of renewable grid integrated hybrid energy storage system. IEEE Trans Ind Electron. 2015;62(12):7728–37. doi:10.1109/TIE.2015.2455063.
6. Saxena A, Shankar R, El-Saadany EF, Kumar M, Zaabi OA, Hosani KA, et al. Intelligent load forecasting and renewable energy integration for enhanced grid reliability. IEEE Trans Ind Appl. 2024;60(6):8403–17. doi:10.1109/TIA.2024.3428956.
7. Liu Y, Li Y, Gooi HB, Jian Y, Xin H, Jiang X, et al. Distributed robust energy management of a multimicrogrid system in the real-time energy market. IEEE Trans Sustain Energy. 2017;10(1):396–406. doi:10.1109/TSTE.2017.2779827.
8. Manikandan M, Saravanan R, Kannayeram G, Saravanan M. Integrating renewable resources and electric vehicles: an approach for effective energy management in DC microgrid. Sol Energy. 2025;299:113775. doi:10.1016/j.solener.2025.113775.
9. Naghibi AF, Akbari E, Shahmoradi S, Pirouzi S, Shahbazi A. Stochastic economic sizing and placement of renewable integrated energy system with combined hydrogen and power technology in the active distribution network. Sci Rep. 2024;14(1):28354. doi:10.1038/s41598-024-79690-7.
10. Emdadi K, Gandomkar M, Nikoukar J. Adaptive robust energy management of smart grid with renewable integrated energy system, fuel cell and electric vehicles stations and renewable distributed generation. Results Eng. 2025;27:106857. doi:10.1016/j.rineng.2025.106857.
11. Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. Deep reinforcement learning: a brief survey. IEEE Signal Process Mag. 2017;34(6):26–38. doi:10.1109/MSP.2017.2743240.
12. Zhao H, Chen H, Yang F, Liu N, Deng H, Cai H, et al. Explainability for large language models: a survey. ACM Trans Intell Syst Technol. 2024;15(2):1–38. doi:10.1145/3639372.
13. Yao T, Xu Y, Wang H, Qiu X, Althoefer K, Qi P. Multi-agent fuzzy reinforcement learning with LLM for cooperative navigation of endovascular robotics. IEEE Trans Fuzzy Syst. 2025. doi:10.1109/TFUZZ.2025.3585934.
14. Zeng T, Badrinarayanan S, Ock J, Lai CK, Barati Farimani A. LLM-guided chemical process optimization with a multi-agent approach. arXiv:2506.20921. 2025.
15. Samarathunga K, Gurusinghe R, Sivasothynathan K, Wanigasekara C, Mars J, Logeeshan V, et al. LLM-guided multi-agent system for natural language-based robot navigation. In: 2025 IEEE World AI IoT Congress (AIIoT). Piscataway, NJ, USA: IEEE; 2025. p. 1055–60.
16. Zhu F, Huang F, Yu Y, Liu G, Huang T. Task offloading with LLM-enhanced multi-agent reinforcement learning in UAV-assisted edge computing. Sensors. 2024;25(1):175. doi:10.3390/s25010175.
17. Li Z, Zhang R, Wang Z, Xie Z, Song Y. LLM-guided decision-making toolkit for multi-agent reinforcement learning. Neurocomputing. 2025;638:130105. doi:10.1016/j.neucom.2025.130105.
18. Joy ER, Bansal RC, Ghenai C, Terzija V, Vorobev P, Kumar R, et al. Artificial intelligence and its applications in renewable integrated power systems. Energy. 2022;256:124696. doi:10.1016/j.energy.2022.124696.
19. Abrar SF, Masood NA, Alam MJ. An adaptive load shedding methodology for renewable integrated power systems. Heliyon. 2024;10(21):e39642. doi:10.1016/j.heliyon.2024.e39642.
20. Heuillet A, Couthouis F, Díaz-Rodríguez N. Explainability in deep reinforcement learning. Knowl Based Syst. 2021;214:106685. doi:10.1016/j.knosys.2020.106685.
21. Thirunavukarasu AJ, Ting DSJ, Elangovan K, Gutierrez L, Tan TF, Ting DSW, et al. Large language models in medicine. Nat Med. 2023;29(8):1930–40. doi:10.1038/s41591-023-02448-8.
22. Kirchenbauer J, Geiping J, Wen Y, Katz J, Miers I, Goldstein T, et al. A watermark for large language models. In: Proceedings of the 40th International Conference on Machine Learning. London, UK: PMLR; 2023. p. 17061–84.
23. Shanahan M. Talking about large language models. Commun ACM. 2024;67(2):68–79. doi:10.1145/3624564.

24. Sun H, Li J. LLM-guided reinforcement learning with representative agents for traffic modeling. arXiv:2511.06260. 2025.
25. Yang F, Liu J, Li K. LLM-guided reinforcement learning for interactive environments. Mathematics. 2025;13(12):1932. doi:10.3390/math13121932.