EV Charging System Considering Power Dispatching Based on Multi-Agent LLMs and CGAN

Zeyuan Niu, *Graduate Student Member, IEEE*, Jiamei Li, Qian Ai, *Senior Member, IEEE*,
Jiamei Jiang, Qiunan Yang, and Haojie Zhou

***Abstract*— The growing adoption of electric vehicles (EVs) has placed significant demands on power grids, necessitating coordination between EV charging and power dispatching. This paper proposes a novel EV charging system using Multi-Agent Large Language Models (LLMs) to enhance recommendations, optimize decision-making, and dynamically adapt to user behaviors and grid conditions. The system includes a User Agent and an EV Charging Station (EVCS) Agent, connected through a Negotiation Platform for secure data sharing. The User Agent provides personalized recommendations based on historical data, while the EVCS Agent adjusts prices in real time using fine-tuned LLMs. A Conditional Generative Adversarial Network (CGAN) model is used to generate user behavior and pricing data to fine-tune the LLMs. The proposed system effectively adapts to dynamic user behaviors and grid conditions by combining Multi-Agent coordination with fine-tuned LLMs and CGAN-generated data. A case study demonstrates the system's ability to balance user preferences with power dispatching, offering scalable, efficient, and intelligent solutions for modern EV ecosystems.**

***Index Terms*— Large language model, electric vehicle, multi-agent, conditional generative adversarial networks, power dispatching, retrieval-augmented generation.**

I. Introduction

THE increasing demand for EV charging puts significant pressure on power grids [1], due to EVs' unpredictability and users' different preferences. It is forecast that, globally by 2035, the number of light-duty EVs and their charging plugs will multiply to over 300 million and 175 million [2], respectively. This large-scale adoption of EVs creates significant challenges for both the power grid and end users [3]. As numerous EVs connect to the grid, the resulting surge in demand can lead to higher peak loads, voltage instability, and congestion, potentially undermining grid reliability. Meanwhile, EV users face additional concerns related to charging costs, time constraints, accessibility of public charging stations, and battery wear. Under these circumstances, there is an urgent need for a multi-perspective approach to effectively coordinate electric vehicles and the power grid.

In the aspect of LLMs' application in EV charging and power dispatching, authors in [4] introduce a user-centric approach to power scheduling using LLMs, leveraging Multi-Agent architectures to interpret user requests and generate optimized resource allocation strategies, specifically applied to EV charging. In [5], authors review an LLM-based agent framework that dynamically simulates and optimizes EV charging behavior by integrating user preferences, psychological characteristics, and environmental factors, enhancing urban charging management efficiency and user satisfaction. An innovative approach for real-time optimal power flow by integrating a GPT-based agent and deep reinforcement learning is presented in [6], enabling the consideration of qualitative, language-based stipulations alongside traditional quantitative objectives. ChatEV, an LLM-based framework designed to predict EV charging demand by transforming charging data into a unified language format is introduced in [7], which enhances urban grid management efficiency and enabling real-time adaptation to various urban scenarios. Reference [8] proposes a hybrid framework that combines a LLM and Dual Deep Deterministic Policy Gradient (D3PG) for power dispatching with electric vehicles in a space-air-ground integrated vehicular network (SAGVN) environment. In this framework, the LLM interprets natural language instructions from EV drivers to guide the D3PG model in generating dispatching strategies that meet user demands. These strategies are then optimized using the Whale Optimization Algorithm (WOA) to enhance dispatching performance and ensure node voltage stability within the network. Ming Jin et al. discuss the application of LLM in automating and democratizing energy management optimization in [9], focusing on how LLM-assisted auto formalism streamlines complex optimization tasks, improves accessibility for non-experts, and enhances the scalability of energy solutions across various scenarios.

In the field of intelligent transportation, the combination of LLM and Multi-Agent is studied in [10], providing an avenue for advancing the capabilities and performance of LLMs through collaboration and knowledge exchange among intelligent agents. Liu et al. explores integrating LLMs with Reconfigurable Intelligent Surfaces (RIS) to enhance communication in the Internet of Vehicles (IoV) in [11]. Traj-LLM, a novel trajectory prediction framework that leverages pre-trained LLMs to infer future vehicular trajectories without explicit prompt engineering is introduced in [12], utilizing sparse context joint encoding and lane-aware probabilistic learning to enhance scene comprehension and social

Received 2 December 2024; revised 9 March 2025; accepted 22 April 2025. This work was supported by the National Key Research and Development Program of China under Grant 2021YFB2401204. The Associate Editor for this article was C. K. Sundarabalan. *(Corresponding author: Jiamei Li.)*

The authors are with the School of Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: lijiamei@sjtu.edu.cn).

Digital Object Identifier 10.1109/TITS.2025.3564389

interactions, ultimately surpassing state-of-the-art methods in most evaluation metrics. A Transformer-based unified framework called VistaGPT for transport automation is proposed in [13] addressing challenges of system and module heterogeneity by integrating diverse vehicular Transformers and utilizing large language models for automated composition, aiming to enhance scalability, flexibility, and the development of intelligent, end-to-end autonomous driving systems. In [14], authors introduce an interpretable, rule-compliant decision-making framework for autonomous vehicles, leveraging Retrieval-Augmented Generation (RAG) and a reasoning module powered by LLMs to retrieve and interpret traffic regulations, enabling adaptive, safe, and region-specific navigation in complex driving scenarios.

Furthermore, research has been conducted on the evaluation and benchmarking of LLMs. ElecBench is introduced in [15] as an evaluation benchmark designed to comprehensively assess the performance of LLMs in the power sector, focusing on metrics such as factuality, logicality, stability, fairness, security, and expressiveness, which are critical for power dispatch applications. LLMs' evaluation for predicting user preferences in recommendation systems is studied in [16] by examining their performance across zero-shot, few-shot, and fine-tuning scenarios, and comparing them with traditional collaborative filtering methods to understand the potential of LLMs in capturing user rating behavior efficiently. Authors in [17] present EvaLLM, a conceptual evaluation stack and platform designed to systematically assess the capabilities of LLMs in generating visualizations, highlighting the framework's multi-layered approach to evaluate syntactic accuracy, representational quality, perceptual design, and semantic insight across various generative tasks.

Currently, the research on the integration of LLM and GAN is emerging. Y Wang .et al introduce LLM-GAN in [18], a framework that combines LLM and adversarial mechanisms to improve explainable fake news detection, addressing challenges such as detecting complex fake news patterns and generating accurate, user-trusted explanations. Leveraging the combined capabilities of LLM and GAN to enhance the robustness and adaptability of recommendation systems by generating diverse and semantically rich prompts, GANPrompt put forward in [19] addresses sensitivity issues and improves performance in dynamic and complex environments. GLEAM, a novel framework that combines GAN and LLM is introduced in [20] to generate realistic adversarial malware samples, leveraging hex code and opcode features to evaluate and enhance the robustness of malware detection systems against evasive threats. Targeted Catastrophic Forgetting (TCF) is presented in [21], combining CGAN and LLM to mitigate privacy risks, specifically Personally Identifiable Information (PII) leakage, while maintaining utility, with experimental validations showing the framework's efficacy in enhancing privacy and contextual coherence under adversarial attacks.

Previous research has explored the applications of LLMs in EV charging and scheduling and the GAN's combination method with LLM. However, while existing studies have examined EV scheduling and EVCS operations, few have comprehensively investigated how EVs and EVCSs can be jointly orchestrated under a Multi-Agent framework to meet dynamic grid dispatch requirements, especially when user preferences and real-time EVCS status are taken into consideration simultaneously. This paper proposes a Multi-Agent-LLM-based charging system that leverages CGAN for generating user and charging station data, enhancing the fine-tuning of the LLMs used in the framework. The system dynamically optimizes EV charging schedules by incorporating power plans, electricity pricing, and user preferences. Through its dual-agent structure, comprising the User Agent and the EVCS Agent, and a Negotiation Platform, the framework enables personalized recommendations for users and adaptive price adjustments by charging stations. By integrating LLMs for advanced decision-making and negotiation and CGAN for robust data generation, this system effectively balances user satisfaction, charging efficiency, and grid load stability.

The main contributions of this paper are as follows:

Multi-Agent-LLM Framework: We propose a novel Multi-Agent-LLM based charging system that integrates the User Agents and the EVCS Agents, interconnected via the Negotiation Platform, to dynamically coordinate EV charging and grid dispatching.

Integration of LLM and CGAN: The system leverages LLMs for advanced decision-making and negotiation processes, while the CGAN-LLM method is employed for data generation, enhancing the system's performance in different scenarios.

Dynamic Charging Optimization: The framework dynamically adjusts EV charging schedules and pricing based on real-time EVCS load predictions and day-ahead load scheme, achieved through a combination of the fine-tuned EVCS Agents' LLMs and Time-LLM based forecasting.

Negotiation Mechanism: A Negotiation Platform is developed to match the user with the related EVCSs. The platform enables users to make informed decisions based on their preferences and the capabilities of the EVCSs, facilitating intelligent price adjustments and enhanced charging efficiency of the EVCSs.

Section II introduces the structure of the Multi-Agent-LLM framework. Section III provides explanation of the model construction. Section IV presents a case study to validate the proposed methodology. Section V concludes the paper with key insights and outlines potential future research directions.

II. Structure Of Multi-Agent-LLM System

The structure of Multi-Agent-LLM based electric vehicle charging system is illustrated in Figure 1. The framework consists of two main parts: the User Agent and the EVCS Agents, interconnected by Negotiation Platform. Compared to traditional centralized scheduling systems [22], [23], our Multi-Agent-LLMs framework offers greater flexibility by enabling real-time adaptation to dynamic user behaviors and grid conditions.

A. User Agent

The User Agent is designed to assist users in identifying optimal charging stations and navigation routes based on their preferences and historical data. The key components are:

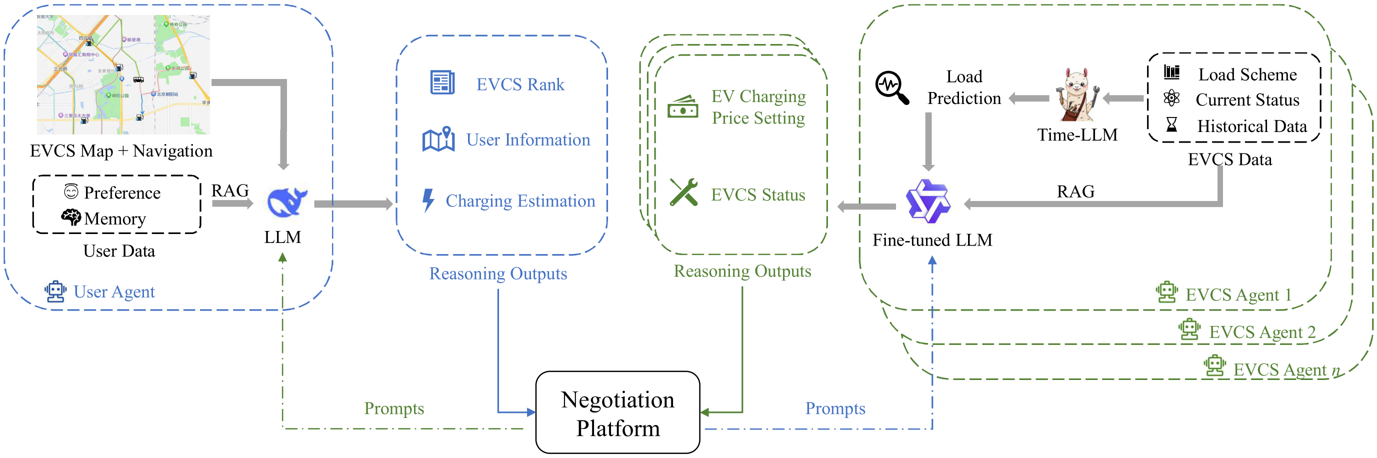


Fig. 1. Structure of Multi-Agent-LLM based EV charging system. The User Agent integrates user-specific data via **RAG (information retrieval)** to provide charging recommendations. The EVCS Agents combine station status and historical data with a Time-LLM for load prediction and a **fine-tuned (modified based on labeled data)** LLM for real-time price setting. The Negotiation Platform mediates data exchanges and pricing negotiations between the User Agent and multiple EVCS Agents, ensuring dynamic coordination of user behaviors and grid conditions.

User Data: This module stores user data, including preferences and past behaviors embedded in prompt by RAG, to create personalized charging recommendations for agents.

EVCS Map & Navigation: Another input is EVCS map & navigation, providing nearby EVCSs location and path for users, which can be accessed by APIs such as Amap-API.

LLM: Given user data and EVCS map & navigation, LLM can reason more detailed EVCS rank, path recommendation and charging estimation.

B. EVCS Agent

The EVCS Agent is responsible for managing EV charging operations, load prediction, and providing the charging price, which components are:

EVCS Data: Data of EVCS Agent consists of historical demand, current status, load prediction and load scheme.

Time-LLM: Due to the abundant text content participating in the system, we introduce the Time-LLM for prediction by combining numerical data and texts. This component uses Time-LLM [24], a time-series adapted version of the LLM, to predict the actual future load at the EVCS, by historical demand and current status.

Fine-tuned LLM: The EVCS Agent uses a fine-tuned LLM to dynamically adjust charging prices and manage station status. The fine-tuning process utilizes a dataset that includes user's past charging prices, past start SOC, and current charging prices, with the most suitable charging price serving as the label. This enables the EVCS Agent to make precise, context-aware pricing adjustments.

C. Negotiation Platform

The Negotiation Platform serves as a communication bridge between the User Agent and the EVCS Agent, matching the specific user with the related EVCSs, enabling filtered and relevant data to pass between the two agents and providing a supervised place for agents to communicate. Unlike traditional one-way communication systems between users and charging stations [25], our Negotiation Platform enables dynamic two-way negotiation, allowing both users and EVCS Agents to adjust charging prices based on real-time data. This platform facilitates an ongoing dialogue between the agents, enhancing both user satisfaction and operational efficiency.

III. Models and Methods

A. Problem Formulation

The goal of this research is to optimize EVCS recommendations and load management using a dual-agent framework with LLMs and RAG capabilities [26], [27]. The User Agent and EVCS Agent collaborate to meet both user-specific charging requirements and grid load balancing needs.

The User Agent seeks to maximize the relevance of EVCS recommendations based on user preferences u and contextual information. Using RAG, it retrieves relevant information $D_u \subseteq D$ conditioned on q_u, and forms a prompt for the LLM:

$$P_\theta(y|q_u, D_u) = \prod_{i=1}^{n} P_\theta(y_i|y_{<i}, q_u, D_u) \tag{1}$$

where $P_\theta(y_i|y_{<i}, q_u, D_u)$ denotes the conditional probability of generating the i-th token given prior tokens, user query q_u, and the retrieved context D_u.

The EVCS Agent aims to balance the charging demand and grid stability by predicting load and adjusting charging prices. Given historical load data h_t and current status s_t, the Time-LLM predicts future load L_{t+1}:

$$L_{t+1} = f_{\text{Time-LLM}}(h_t, s_t) \tag{2}$$

The LLM in the EVCS Agent is fine-tuned to enable it to set charging prices for electric vehicles. While general-purpose language models have broad knowledge and reasoning capabilities, they lack the specialized understanding needed to dynamically set prices based on factors like load predictions, energy demand, user preferences, and historical pricing data. By fine-tuning, the LLM can learn these domain-specific patterns, making it capable of generating accurate and responsive pricing decisions that align with the operational needs of electric vehicle charging stations. This tailored functionality enhances the LLM's utility, ensuring it provides contextually relevant and practical pricing solutions. In this paper, the pre-trained LLM is fine-tuned using the LoRA method [28]. LoRA can be regarded as a special type of Adapter, with the core idea of reducing the number of parameters through low-rank decomposition. The introduced parameter changes are implicitly contained within the decomposition matrices.

Suppose in a particular layer of the model, the original weight matrix is denoted as W. In the LoRA fine-tuning process, we introduce two low-rank matrices A and B to approximate the adjustment to W:

$$W_{\text{LoRA}} = W + \Delta W \tag{3}$$

$$\Delta W = A \times B \tag{4}$$

where $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times d}$ are two low-rank matrices, with $r \ll d$, which means the rank of A and B is much smaller than the rank of W. A and B are the only parameters trained during fine-tuning, while W remains frozen.

The result of LoRA fine-tuning is an adapted set of parameters θ_{LoRA}:

$$\theta_{\text{LoRA}} = \theta + \Delta\theta \tag{5}$$

$$\Delta\theta = \{\Delta W_1, \Delta W_2, \ldots, \Delta W_n\} \tag{6}$$

where n represents the number of layers participating in the fine-tuning process.

During LoRA fine-tuning, we learn the values of A and B by minimizing the loss for the specific generation or decision-making task. The loss function can be represented as follows:

$$\mathcal{L}(\theta_{\text{LoRA}}) = \sum_i \log P_{\theta_{\text{LoRA}}}(y_i|y_{<i}, q_u, D_u) \tag{7}$$

In LoRA fine-tuning, the original model parameters θ are frozen, and only the parameters of the low-rank matrices A and B are trainable. By minimizing $\mathcal{L}(\theta_{\text{LoRA}})$, the model can update the values of A and B, thereby indirectly adjusting the parameters to better adapt the model to the specific task. Thus, the reasoning function can be described as:

$$P_{\theta_{\text{LoRA}}}(y|q_u, D_u) = \prod_{i=1}^{n} P_{\theta_{\text{LoRA}}}(y_i|y_{<i}, q_u, D_u) \tag{8}$$

The multi-objective optimization problem can be abstracted as follows:

The objective of the User Agent is to recommend the most suitable EVCSs for the user.

$$\max_x \sum_{j=1}^{M} R_{ij} \cdot x_j \tag{9}$$

$$\text{s.t.} \quad \sum_{j=1}^{M} x_j = 3 \tag{10}$$

$$x_j = \begin{cases} 1, \text{if EVCS}_j \text{is chosen} \\ 0, \text{otherwise} \end{cases} \tag{11}$$

$$R_{ij} = f_{user-agent}(i, j) \tag{12}$$

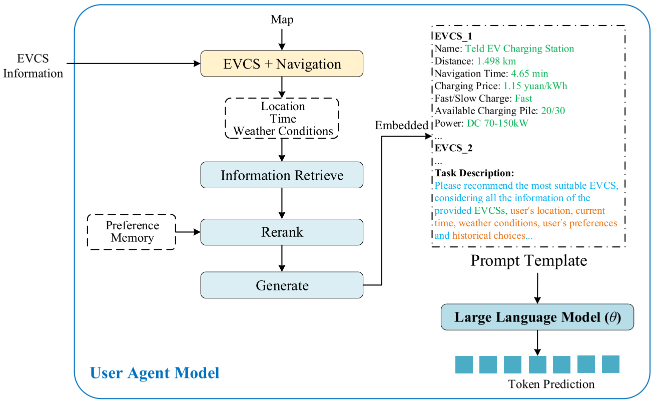

Map
EVCS Information
EVCS + Navigation
Location
Time
Weather Conditions
Information Retrieve
Preference Memory
Rerank
Generate
Embedded
EVCS_1
Name: Teld EV Charging Station
Distance: 1.498 km
Navigation Time: 4.65 min
Charging Price: 1.15 yuan/kWh
Fast/Slow Charge: Fast
Available Charging Pile: 20/30
Power: DC 70-150kW
...
EVCS_2
...
Task Description:
Please recommend the most suitable EVCS, considering all the information of the provided EVCSs, user's location, current time, weather conditions, user's preferences and historical choices...
Prompt Template
Large Language Model (θ)
Token Prediction
User Agent Model

Fig. 2. Model of User Agent.

where $f_{user-agent}(i,j)$ represents the User Agent Model's result of user i's rating towards EVCS j. M is the total number of EVCSs.

The objective of the EVCS Agent is to gain as much revenue as possible while aligning the load curve to the load scheme.

$$\max_p f_{evcs-agent}(J_1, J_2) \tag{13}$$

$$\text{s.t.} \quad p_{i,t} \in [0.8r_t, 1.2r_t] \tag{14}$$

$$J_1 = \sum_{i=1}^{N} (\int_t (p_{i,t} \cdot P_{i,t})\Delta t) \tag{15}$$

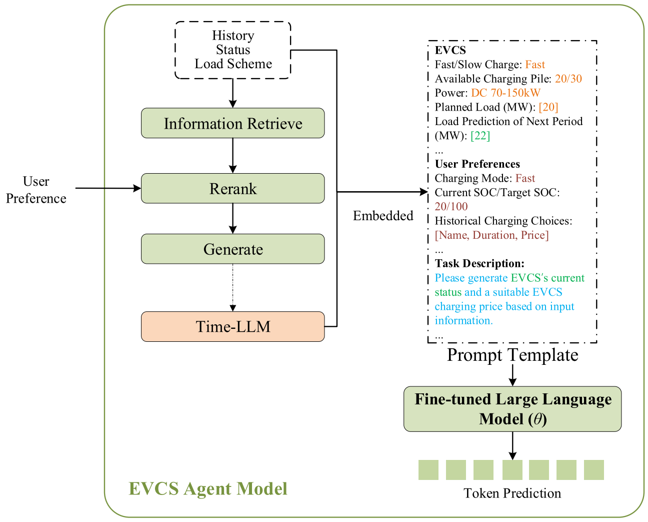
$$J_2 = 1 - \frac{\sum_t \left|\overline{P}_{r,t} - P_{s,t}\right|}{\sum_t \left|P_{s,t}\right|} \tag{16}$$

where $f_{evcs-agent}(J_1, J_2)$ represents the EVCS Agent Model's results of judging and weighting the impact of J_1 and J_2, r_t represents the normal charging price at time t, $p_{i,t}$ is the charging price of user i at time t provided by the EVCS Agent, $P_{i,t}$ is the charging power of user i at time t, $\bar{P}_{r,t}$ is the hour-average charging power of the EVCS at time t, and $P_{s,t}$ is the load scheme of the EVCS at time t.

Although the optimization problem is formulated, the realization of $f_{user-agent}(i,j)$ and $f_{evcs-agent}(J_1, J_2)$ is not specified and shall be given and calculated by the LLMs and Agents put forward in the following sections. The principal rationale for the formulation of this problem is to elucidate the decision variable and the ultimate objective.

B. User Agent Model

The User Agent Model illustrated in Figure 2 is designed to provide optimized EVCS recommendations by leveraging a combination of location-based data and an LLM. The process begins with the *EVCS + Navigation* module, which gathers user location, time, and weather conditions context through the Amap-API and integrates relevant EVCS information. The Information Retrieval module then extracts pertinent data points based on these inputs. Next, the *Rerank* module [29] organizes the retrieved information, prioritizing the most relevant EVCS options according to factors such as proximity, availability, and user preferences. The *Generate* module utilizes a prompt template, combining the user query with the



EVCS
Fast/Slow Charge: Fast
Available Charging Pile: 20/30
Power: DC 70-150kW
Planned Load (MW): [20]
Load Prediction of Next Period (MW): [22]
...
User Preferences
Charging Mode: Fast
Current SOC/Target SOC: 20/100
Historical Charging Choices: [Name, Duration, Price]
...
Task Description:
Please generate EVCS's current status and a suitable EVCS charging price based on input information.
...

Fig. 3. Model of EVCS Agent.

reranked EVCS information and specific task instructions, to structure an input for the LLM. This model operates on a token-based prediction mechanism $P_\theta(y_i|y_{<i}, q_u, D_u)$ to produce coherent and contextually appropriate responses.

The whole User Agent Model is described as $f_{user-agent}(i,j)$, as mentioned in (11) and the *rating* is concealed within the process of LLM, outputting only the decision variable x.

C. EVCS Agent Model

The EVCS Agent Model depicted in Figure 3 is designed to manage EVCS recommendations by leveraging historical data, station status, and load scheduling information in combination with a *Time-LLM* based predictions. The process begins with the *Information Retrieval* module, which gathers relevant data on charging station availability and current load conditions. Subsequently, the *Rerank* module prioritizes the retrieved information based on user preferences as extra input. The *Generate* module then utilizes this prioritized data to construct a structured Prompt Template that serves as input to the fine-tuned LLM. This prompt includes detailed information on charging stations, load conditions, and operational parameters, enabling the fine-tuned LLM to generate accurate and personalized EVCS recommendations that are specifically tailored for this task. Additionally, the *Time-LLM* module predicts future load conditions L_{t+1}, feeding this prediction into the generation pipeline to adjust recommendations dynamically based on anticipated demand. The fine-tuned LLM outputs the next token prediction $P_{\theta_{\text{LoRA}}}(y_i|y_{<i}, q_u, D_u)$, creating a coherent response that aligns with user preferences and grid requirements.

D. Negotiation Platform Model

Figure 4 illustrates the model of the Negotiation Platform designed to facilitate dynamic and privacy-preserving interactions between a User Agent and multiple EVCS Agents by matching them together.

The Negotiation Platform initially collects information from both the user and the EVCSs. Subsequently, the *Priority Rank* module, powered by the User Agent, evaluates and prioritizes the gathered EVCSs' data based on the user's information and match the user with the related EVCSs. Finally, the *Privacy Filter* module safeguards sensitive data before it is shared across agents, yielding Filtered EVCS Information and Filtered User Information.

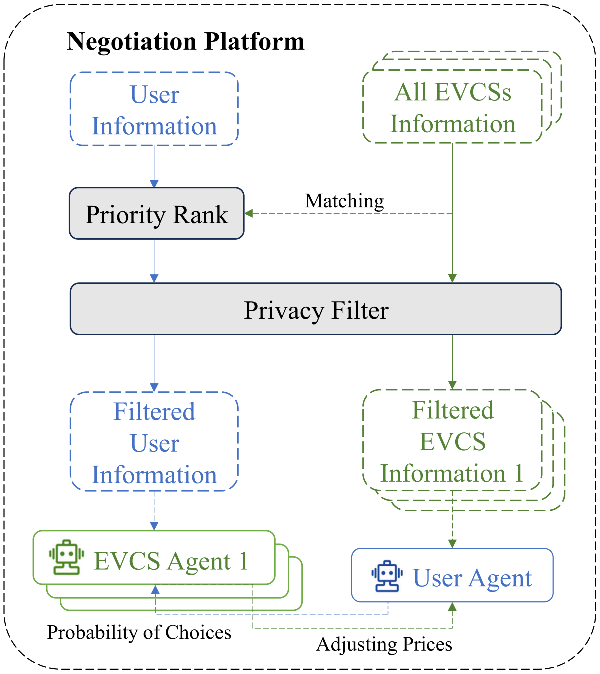
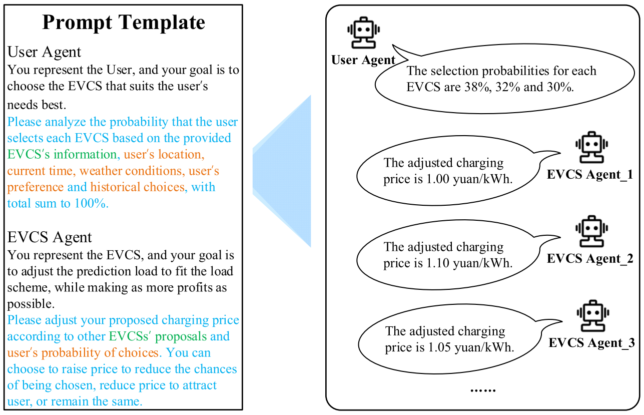


Fig. 4. Model of negotiation platform.



Prompt Template

User Agent
You represent the User, and your goal is to choose the EVCS that suits the user's needs best.
Please analyze the probability that the user selects each EVCS based on the provided EVCS's information, user's location, current time, weather conditions, user's preference and historical choices, with total sum to 100%.

EVCS Agent
You represent the EVCS, and your goal is to adjust the prediction load to fit the load scheme, while making as more profits as possible.
Please adjust your proposed charging price according to other EVCSs' proposals and user's probability of choices. You can choose to raise price to reduce the chances of being chosen, reduce price to attract user, or remain the same.

User Agent
The selection probabilities for each EVCS are 38%, 32% and 30%.
The adjusted charging price is 1.00 yuan/kWh.
EVCS Agent_1
The adjusted charging price is 1.10 yuan/kWh.
EVCS Agent_2
The adjusted charging price is 1.05 yuan/kWh.
EVCS Agent_3
......

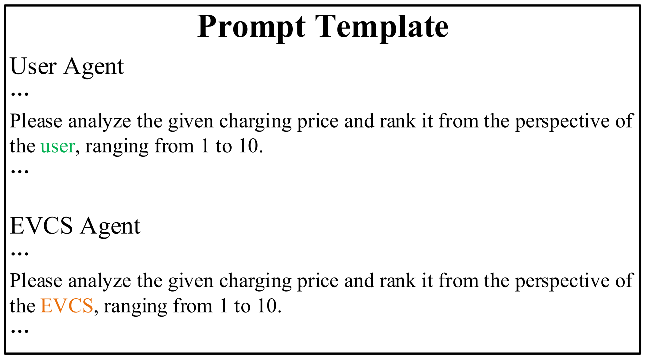
Fig. 5. Template and example of negotiation platform.

The User Agent receives the filtered EVCS information and generates a set of possibilities of choices, which it communicates to the EVCS Agents. Each EVCS Agent, in turn, utilizes the filtered user information to tailor its response, engaging in a negotiation process to determine optimal charging prices. The adjusted charging prices are then provided back to the User Agent, concluding the interaction.

The prompt template used in negotiating and an example of the negotiation's process are shown in Figure 5.

E. Data Generation Model

In this part, we use a CGAN-LLM based model [30] [31] to generate the case's data and the dataset used in fine-tuning.



Prompt Template

User Agent

...

Please analyze the given charging price and rank it from the perspective of the user, ranging from 1 to 10.

...

EVCS Agent

...

Please analyze the given charging price and rank it from the perspective of the EVCS, ranging from 1 to 10.

...

Fig. 6. Prompt used for rating.

It is important to note that the CGAN model may encounter potential limitations such as overfitting and data bias. To address these issues, the LLM is incorporated into the reward function to imitate human decision, and a broad range of user behaviors is simulated, ensuring that the dataset reflects diverse real-world scenarios.

The task involves a generator $G(z, c)$ producing synthetic data x_{fake} conditioned on c, which represents domain-specific auxiliary information, and random noise z. The discriminator $D(x, c)$ evaluates the realism and conditional consistency of the generated data, while the LLM provides additional semantic rewards based on context.

$$x_{fake} = G(z, c) \tag{17}$$

The reward function can be described as:

$$R = R_D + \lambda \cdot R_{LLM} \tag{18}$$

$$R_D = \log D(G(z, c), c) \tag{19}$$

$$R_{LLM} = Parse(f_{User}(x, s)) + Parse(f_{EVCS}(x, s)) \tag{20}$$

where R_D is the adversarial reward from the discriminator, R_{LLM} is the LLM-generated semantic reward, and λ is a hyperparameter controlling the influence of the LLM reward. In R_{LLM}, x is the generated charging price and s is the current state. f_{User} and f_{EVCS} indicate the rating results from the User Agent and the EVCS Agent (currently un-fine-tuned) respectively, which are generated from the prompt template shown in Figure 6. The reason why they are different from (9) and (13) is that the prompt template used is not the same. *Parse*() is the function that extracts the numerical reward from the LLM's response.

The Generator's loss function L_G and the Discriminator's loss function L_D can be described as follows:

$$\begin{aligned} L_G = &-\mathbb{E}_{z \sim p_z(z), c \sim p_c(c)}[\log D(G(z, c), c) \\ &+ \lambda \cdot f_{\text{LLM}}(G(z, c), s)] \end{aligned} \tag{21}$$

$$\begin{aligned} L_D = &-\mathbb{E}_{x_{\text{real}} \sim p_{\text{data}}(x), c \sim p_c(c)}[\log D(x_{\text{real}}, c)] \\ &- \mathbb{E}_{z \sim p_z(z), c \sim p_c(c)}[\log(1 - D(G(z, c), c))] \end{aligned} \tag{22}$$

The combined objective function is:

$$\begin{aligned} \min_G \max_D \; &\mathbb{E}_{x_{\text{real}} \sim p_{\text{data}}(x), c \sim p_c(c)}[\log D(x_{\text{real}}, c)] \\ &+ \mathbb{E}_{z \sim p_z(z), c \sim p_c(c)}[\log(1 - D(G(z, c), c))] \\ &+ \lambda \cdot f_{\text{LLM}}(G(z, c), s) \end{aligned} \tag{23}$$

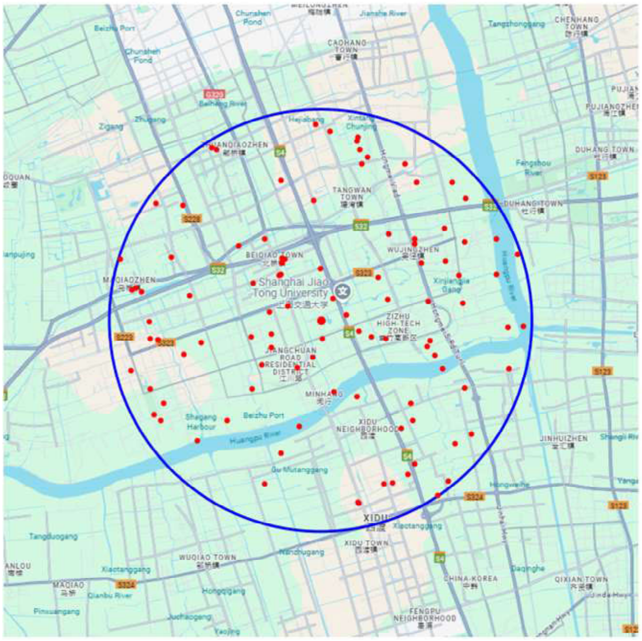


Fig. 7. Example of users' generation.

where $x_{\text{real}} \sim p_{\text{data}}(x)$ is the real data samples drawn from the data distribution, $z \sim p_z(z)$ is the random noise input for the generator, usually from a Gaussian or uniform distribution, and $c \sim p_c(c)$ is the condition vector sampled from the condition distribution.

IV. Case Study

A. Case Introduction and Initialization

A total of 100 users with different charging needs are randomly generated and situated in a 6 km radius near (31.0254° N, 121.4338° E) per hour, as shown in Figure 7. Whether the user's location is on the road does not have a significant influence on the navigation. The initial SOC range is set at [5], [30], and the users exhibited diverse preferences. The charging price is restricted to range in $[0.8, 1.2]{*}r_t$. It needs to be explained that p_t is set every hour and does not change during each user's charging period.

The generated users can be divided into different types, according to their behavior and preferences. The major distinctions of user's preferences are shown in Table I.

However, user's preferences and historical data do not strictly conform to the types. Noises are added to test the system's robustness.

The LLMs used in the case study and relevant parameters' settings are shown in Table II.

The dataset used for fine-tuning has a template, shown in Figure 8. The suitable price is generated from the CGAN-LLM model, and the normalized results of generation are shown in Figure 9. The loss functions' changes during the training process are shown in Appendix Figure 19.

The Time-LLM is trained and tested on private dataset and the results of Time-LLM's results on test data set is shown in Figure 20.

The dataset consists of three parts: *Instruction*, *Input* and *Output*. *Instruction* is the objective description for LLM to

Dataset Template

"instruction": "Generate a suitable EVCS charging price based on input information.
...",

"input": "planned_load (MW): 1.30,
next_period_predicted_load (MW): 1.50,
estimated_start_charging_time: 10:53,
user_preference: Any,
current_soc: 14,
user_past_charging_prices (yuan/kWh): [1.01, 1.17, 0.98, 0.8, 1.08, 1.22, 0.91, 1.26, 1.27, 1.07],
average_charging_price_at_present_time (yuan/kWh): 1.25
user_past_start_soc: [5, 15, 23, 22, 28, 23, 26, 8, 27, 15]",

"output": "The suitable charging price for the user is 1.15 yuan/kWh."

Fig. 8. Example of the dataset.

TABLE I
MAJOR DISTINCTIONS OF USER'S PREFERENCES

Preferences	Content	Value	Proportion
Start Charging Time	Morning	6:01-12:00	20%
	Noon	12:01-18:00	20%
	Evening	18:00-24:00	20%
	Midnight	0:01-6:00	20%
	Any	/	20%
Charging Mode	Fast	/	70%
	Slow	/	10%
	Any	/	20%
Charging Price	Low	$[0.8,1.0]*r_t$	50%
	High	$[1.0,1.2]*r_t$	20%
	Any	/	30%
Start SOC	Low	[5,15]	30%
	High	[16,30]	30%
	Any	/	40%

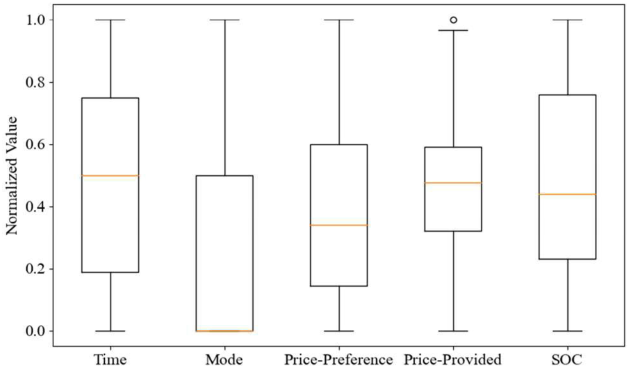


Fig. 9. Normalized results of generation.

TABLE II
MODELS AND PARAMETERS

Category	Parameter	Value
Choice of basic LLM used	Model	Deepseek-chat/ GPT-4o mini
	Fine-tuned LLM	Qwen7B-chat
Parameters for LLM's reasoning	Temperature	0.2
	Top P	0.75
	Frequency penalty	0
Model used in reranking	Rerank-Model	gte-rerank
	Top K	3
Max negotiation iteration	*n*	4
Parameters in fine-tuning	Epoch	10
	Max token length	1024
	LoRA rank	8
EV(CS) settings	SOC when leaving	100
	Charging power (if not confirmed)	100kW (fast) 7kW (slow)
	EV battery capacity	100kWh
	Users per hour/EVCSs' total capability	100/200
Time-LLM hyperparameters	Seq_len	72
	Pred_len	24
	Model	GPT-2
	Patch	16
	LLM_dim	768
	LLM_layers	4
CGAN hyperparameters	Epochs	5000
	Batch_size	64
	Condition dim	8
	Learning rate	0.0001 (D) 0.0002 (G)

understand user's intention. *Input* contains the information needed for LLM to understand the generation process of charging price. *Planned load* and *next period prediction load* is used to indicate the basic goal of setting charging price, which is to either set the price lower to attract the user to charge in this specific EVCS or to set the price higher to "prevent" the user from charging in this EVCS. *Estimated start charging time* and *average charging price at present time* are mainly used to set the datum line for price's setting. *User preferences* show the user's preferred charging mode. *Current soc* and *user past start soc* reflect the user's urgency to charging. *User past charging price* is presented to show the user's sensitivity to charging prices. The above description is also added to the *instruction.*

The flow of the simulation is shown in Figure 10.

1) Generate Users and EVCSs and initialize all agents with basic information.
2) Pass the information of EVCSs and Users through Negotiation Platform and the User Agent is matched with the EVCS Agents.

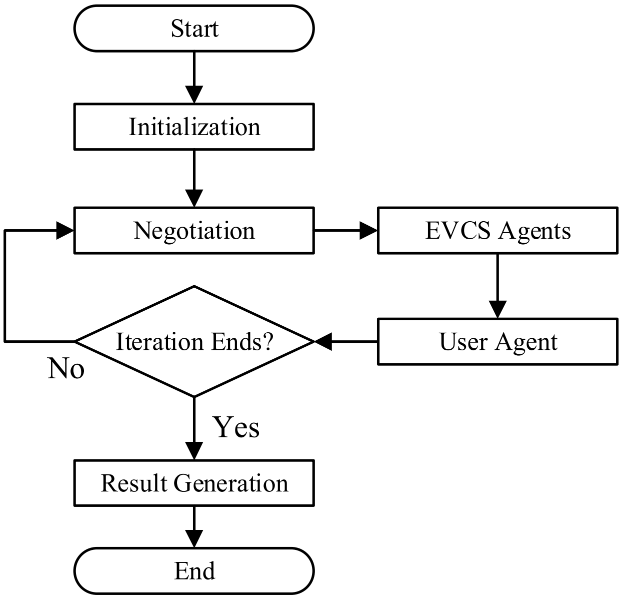
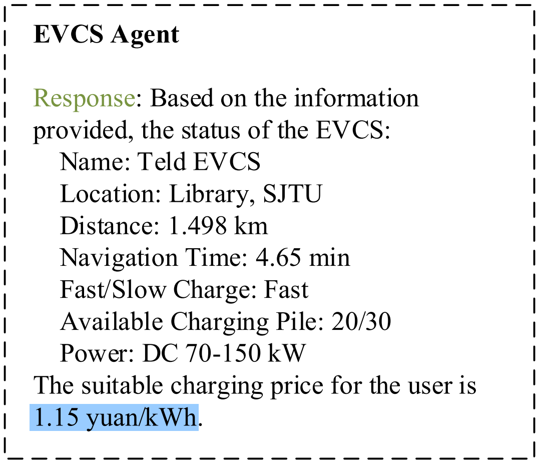


Fig. 10. Flow of the simulation.



EVCS Agent

Response: Based on the information provided, the status of the EVCS:

Name: Teld EVCS
Location: Library, SJTU
Distance: 1.498 km
Navigation Time: 4.65 min
Fast/Slow Charge: Fast
Available Charging Pile: 20/30
Power: DC 70-150 kW

The suitable charging price for the user is 1.15 yuan/kWh.

Fig. 11. Reasoning result of EVCS Agent.

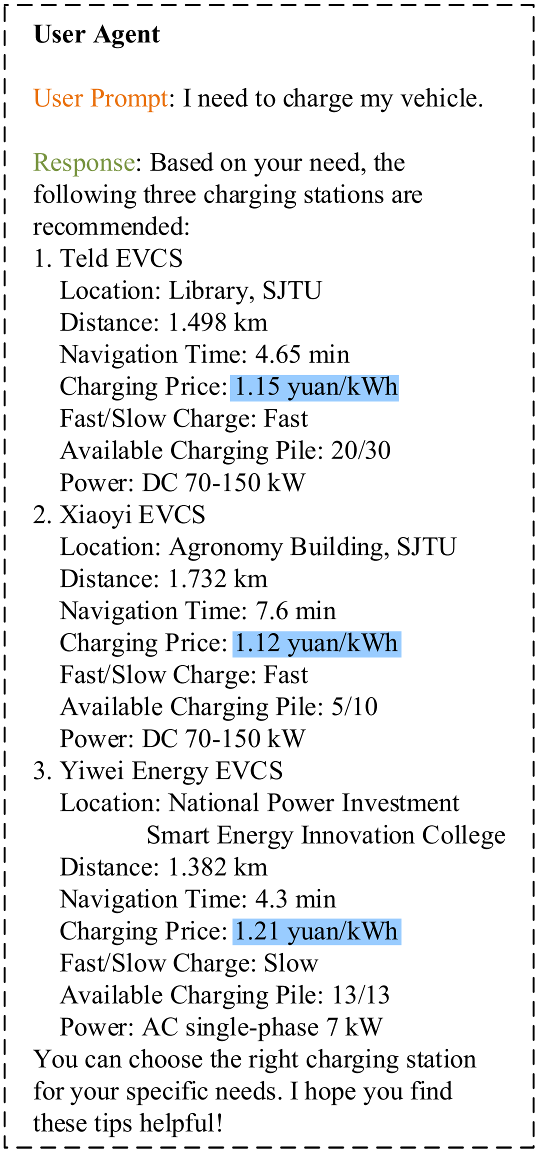
3) EVCS Agents and User Agent start to negotiate on the setting of charging price.
4) Generate the final results.

B. Single User Simulation Results

In this section, the data used is collected from true world at real time rather than generation. The *User*'s location is exactly (31.0254° N, 121.4338° E) and the corresponding data can be seen in Figure 7.

Figure 11 and Figure 12 illustrate the simulation results of the EVCS Agent and the User Agent, respectively. The User Agent, in response to a prompt, recommends multiple nearby charging stations, displaying essential details for each option. The EVCS Agent provides detailed information of specific station, including location, navigation time, charging speed, power capacity, and a calculated price per kWh.

The user's past 10 chosen EVCSs and average accepted charging price are shown in Figure 13.



User Agent

User Prompt: I need to charge my vehicle.

Response: Based on your need, the following three charging stations are recommended:

1. Teld EVCS
   Location: Library, SJTU
   Distance: 1.498 km
   Navigation Time: 4.65 min
   Charging Price: 1.15 yuan/kWh
   Fast/Slow Charge: Fast
   Available Charging Pile: 20/30
   Power: DC 70-150 kW
2. Xiaoyi EVCS
   Location: Agronomy Building, SJTU
   Distance: 1.732 km
   Navigation Time: 7.6 min
   Charging Price: 1.12 yuan/kWh
   Fast/Slow Charge: Fast
   Available Charging Pile: 5/10
   Power: DC 70-150 kW
3. Yiwei Energy EVCS
   Location: National Power Investment Smart Energy Innovation College
   Distance: 1.382 km
   Navigation Time: 4.3 min
   Charging Price: 1.21 yuan/kWh
   Fast/Slow Charge: Slow
   Available Charging Pile: 13/13
   Power: AC single-phase 7 kW

You can choose the right charging station for your specific needs. I hope you find these tips helpful!

Fig. 12. Reasoning result of User Agent.

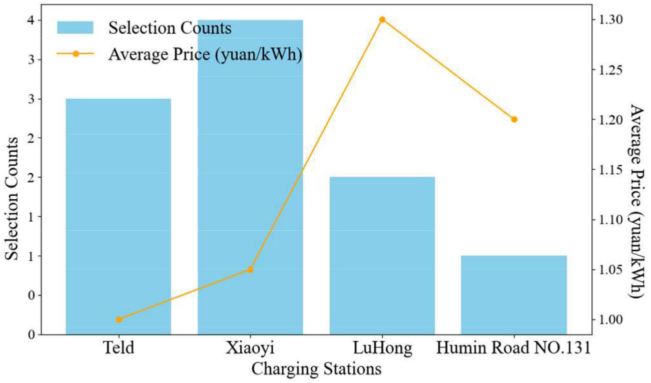
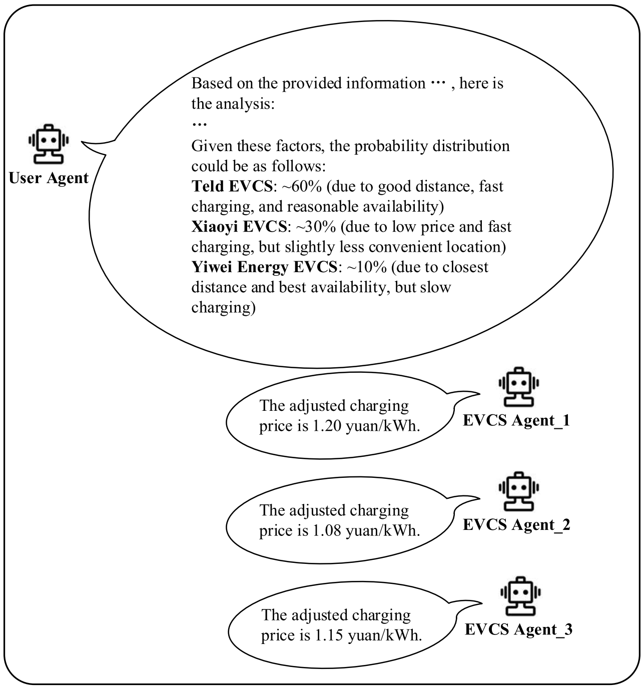


Fig. 13. The user's historical choices.

The impact of User's preference, historical data and weather conditions on the Agent's final results is studied and shown in Table III. The LLM used is GPT-4o-mini, the discussion of the LLM evaluation feature can be referred in Table IX.

TABLE III
IMPACT OF USER'S PREFERENCE, HISTORICAL DATA AND WEATHER CONDITIONS ON THE AGENT'S FINAL RESULTS

	Choice	Agent Result
Prefernce (Historical Data= Not Considered) (Weather Conditions= Sunny)	Fast	1. Teld (fast) 2. Xiaoyi (fast) 3. Humin Road NO.131 (fast)
	Slow	1. Yiwei (slow) 2. StarCharge-Baijinhan (slow) 3. Joy City (slow)
	Any	1. Teld (fast) 2. Xiaoyi (fast) 3. Yiwei (slow)
Historical Data (Preference=Any) (Weather Conditions= Sunny)	Not Considered	1. Teld (fast) 2. Xiaoyi (fast) 3. Yiwei (slow)
	Considered	1. Xiaoyi (fast) 2. Teld (fast) 3. LuHong (fast)
Weather Conditions (Preference=Any) (Historical Data= Not Considered)	Sunny	1. Teld (fast) 2. Xiaoyi (fast) 3. Yiwei (slow)
	Cloudy	1. Teld (fast) 2. Xiaoyi (fast) 3. Yiwei (slow)
	Rainy	1. Teld (fast) 2. Yiwei (slow) 3. Joy City (slow)



Based on the provided information ⋯ , here is the analysis:
…
Given these factors, the probability distribution could be as follows:
Teld EVCS: ~60% (due to good distance, fast charging, and reasonable availability)
Xiaoyi EVCS: ~30% (due to low price and fast charging, but slightly less convenient location)
Yiwei Energy EVCS: ~10% (due to closest distance and best availability, but slow charging)

The adjusted charging price is 1.20 yuan/kWh.

The adjusted charging price is 1.08 yuan/kWh.

The adjusted charging price is 1.15 yuan/kWh.

Fig. 14. Conversation among agents.

From the table, user's preferences strongly impact the agent's recommendations: fast-charging options (e.g., Teld) surface when speed is preferred, while slower stations (e.g., Yiwei) cater to those favoring slower charging. Once historical data is considered, prior visits (e.g., Xiaoyi) carry more weight. Meanwhile, weather conditions further refine choices. On sunny or cloudy days, weather conditions have a minor impact on the agent's recommendations. However, during rainy weather, the agent prioritizes nearby stations to minimize travel distance.

C. Negotiation Simulation Results

To introduce the negotiation among agents, we will use the user and its scenario from *B. Single User Simulation Results*. The conversation is shown in Figure 14. The User Agent would put forward some analysis based on the perspective of the user and give the rational results of probabilities of user's choosing each EVCS. After that, the EVCS Agents would respond to the given probabilities and adjust their provided charging prices independently, according to their own analysis on the situation, in which they must take both the user and the EVCS's status into consideration to generate a rational and effective response.

The negotiation's background is: Teld with the planned load smaller than the predicted load, Xiaoyi with the planned load almost the same as the predicted load, and Yiwei with the planned load larger than the predicted load. The results of the negotiation can be seen in Table IV, from which we can tell that Agents are smart enough to understand the competition situation and adjust dynamically according to the user and the environment. When we look at the results of the second iteration, we find that Rank 1st can sometime raise the price to either meet the need of load deduction or to gain more charging fee; Rank 2nd is likely to maintain pretty much the same, which is probably out of the conservative decision-making habits; Rank 3rd, with perceiving the fact that it can hardly be chosen, would take radical tactics to decrease the charging price, if needed. Agents' different behaviors, occasionally, influence the second iteration enough to re-rank the agents' chosen probabilities.

When switching from DeepSeek-Chat to GPT-4o-mini, distinct outcomes are observed in the negotiation process, seen in Table IV & V. This variation leads each LLM to weigh user preferences and station status differently. For instance, GPT-4o-mini employs a self-devised probability model that may prioritize lower prices to secure a higher user selection rate, while DeepSeek-Chat tends to adjust prices more conservatively based on other contextual cues. These divergent behaviors highlight the adaptability of our proposed framework: it seamlessly accommodates different language models and preserves the Multi-Agent negotiation mechanism's ability to balance user needs with grid and station objectives.

In the process of negotiation, GPT-4o-mini utilize a self-made model to calculate the probabilities rather than simply analyze them. The model put forward by GPT-4o-mini is shown in Appendix Table VII. As the process of iteration progresses, Yiwei is most likely to emerge triumphant by reducing the price of charging the most, while Teld raise the price due to its superfluous load. Xiaoyi does not show major adjustment. It can be posited that as the number of iterations increases, the probability will remain relatively consistent. The alteration of prices and probabilities is indicative of both the user's preference and the EVCSs' requirements, thereby substantiating the efficacy of our methodology.

D. Multiple Users Simulation Results

In this part we use Teld EVCS (fast) at SJTU and Yiwei (slow) as examples to show the influence of our model on

TABLE IV
PROCESS OF NEGOTIATION: DEEPSEEK-CHAT

Rank	Name	Iteration 1		Iteration 2		Iteration 3		Iteration 4	
		Price	Probability	Price	Probability	Price	Probability	Price	Probability
1	Teld	1.15	60%	1.20	40%	1.15	40%	1.10	40%
2	Xiaoyi	1.12	30%	1.08	50%	1.05	50%	1.00	50%
3	Yiwei	1.21	10%	1.15	10%	1.05	10%	1.00	10%

TABLE V
PROCESS OF NEGOTIATION: GPT-4O-MINI

Rank	Name	Iteration 1		Iteration 2		Iteration 3		Iteration 4	
		Price	Probability	Price	Probability	Price	Probability	Price	Probability
1	Teld	1.15	37%	1.17	34%	1.18	32%	1.19	31%
2	Xiaoyi	1.12	35%	1.12	34%	1.11	33%	1.11	34%
3	Yiwei	1.21	28%	1.18	33%	1.16	35%	1.15	35%

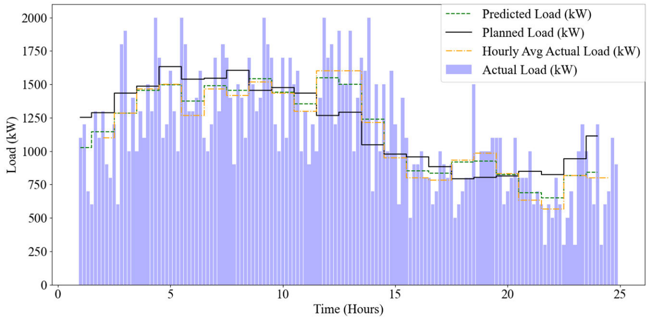


Fig. 15. Load curves of Teld EVCS.

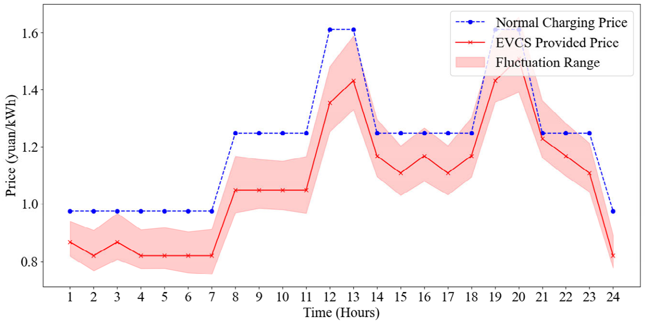


Fig. 16. Comparison between normal charging price and the provided charging price (Teld).

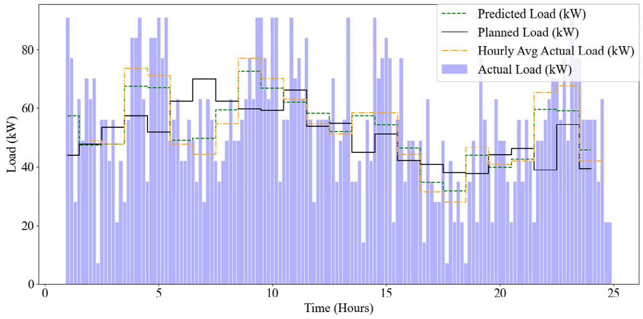


Fig. 17. Load curves of Yiwei EVCS.

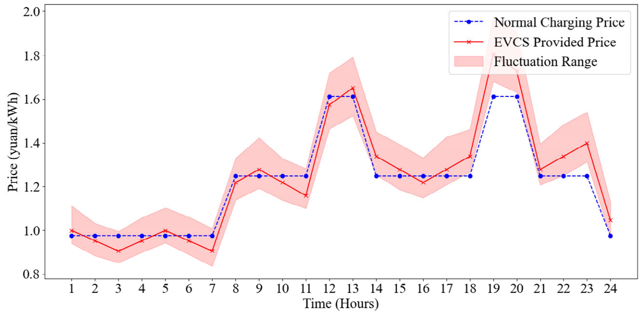


Fig. 18. Comparison between normal charging price and the provided charging price (Yiwei).

power dispatching. The probability of each user selecting an EVCS is calculated, and the resulting behavior is modelled to determine the impact on load.

The load curves of Teld are shown in Figure 15, from which we can tell that the proposed model make the actual load curve more adjacent to the load scheme, which reduces the error that occurs in the prediction of the load. In instances where the predicted load is less than the load scheme, the EVCS Agent will reduce the price in order to attract users. This will consequently result in an increase in the actual load, thereby meeting the requirement. The planned load is manually set.

Figure 16 shows the normal charging price and the charging price provided by EVCS Agent. Due to the quick turnaround and relatively large capacity of fast-charging stations, Teld can offer an average charging price lower than the normal rate. This strategy attracts more users and thus enables Teld to meet its planned load target. However, from another perspective, this also means the planned load is slightly higher than what Teld's demand would normally be without the proposed system. Given this, determining a reasonable planned load is a separate issue and is beyond the scope of this paper.

Figure 17 and Figure 18 illustrate the results of Yiwei, a slow EVCS. Compared to Teld, the fast EVCS, Yiwei's ability in controlling its load to fit the planned load is weaker, due to its smaller capacity and the longer time EVs spend charging there. Meanwhile, the power of users charging also has an impact when the EVCS faces a sudden load change, where Teld's performance is better as well.

TABLE VI
COMPARISON OF METHODS

Method	**Average Charging Price (yuan/kWh)**	**EVCS Revenue (yuan)**	**Average Load Aligning Rate**	**Overall Score**
Teld				
RL-based Model	1.16	**30380.98**	77.82%	97.9
Our Method	**1.12**	28752.34	**80.10%**	**98.2**
Optimization Model	1.13	29065.58	70.45%	94.2
Yiwei				
RL-based Model	1.36	**1618.56**	75.21%	96.1
Our Method	**1.24**	1452.39	**77.37%**	**96.6**
Optimization Model	1.30	1532.24	64.87%	91.3

When we look at the charging price curve provided by two EVCSs, Yiwei has an average higher price for users, which seems to be abnormal. However, this is reasonable since slow charging takes up the charging pile for longer time duration, resulting in the seller's market for those users whose preference is slow charging.

E. Comparison of Proposed Method With Others

A reinforcement learning method for real-time pricing and scheduling control in EVCS is proposed in [32]. We use Average Charging Price, EVCS Revenue, and Average Load Aligning Rate. The optimization model can be referred in *Section III*. The experiment is run on Teld and Yiwei and the results are shown in Table VI. The setting of the overall score is shown in Appendix (A2).

The RL-based method has been demonstrated to achieve the highest revenue across both operators, making it an optimal choice for revenue maximization. However, the method proposed in this paper offers a superior trade-off between revenue and load alignment, which is essential for maintaining operational stability and user satisfaction. While the Optimization Model is conceptually straightforward, it has been observed to face challenges in effectively handling dynamic load conditions, particularly in achieving high load alignment rates. These findings underscore the benefits of RL-based and hybrid approaches in addressing the complexities of EVCS operations.

V. CONCLUSION

This paper presents an EV charging system considering power dispatching based on Multi-Agent-LLM. The Multi-Agent-LLM model comprises three distinct models: the User Agent, the EVCS Agent and the Negotiation Platform. The User Agent represents the user within the system, providing information and recommendations regarding the EVCS to satisfy the user's diverse charging requirements and preferences. The EVCS Agent represents the EVCS in the system. Its primary objective is to adjust the charging price based on the EVCS status and part of the user's information, thereby enabling the dispatching of power. The Negotiation Platform is designed to facilitate discussions between all Agents on the setting of charging prices, thereby simulating the situation in which a user is faced with multiple potential choices of EVCSs.

In the case study, we employ a LLM-CGAN combined model to emulate the decision-making processes of both users and EVCSs. The objective is to generate the data pertaining to users and the dataset utilized in the LLM's fine-tuning. In the single simulation section, the focus is on a single user, with the results of the agents presented and an analysis of the impact of the user's history on the outcomes. The negotiation simulation component presents and analyses the negotiation process undertaken by the agents. In the multiple users simulation section, the impact of a large number of EVs on the load dispatching of the EVCSs is investigated and analyzed. The findings demonstrate that the proposed methodology is capable of addressing the user's requirements and preferences, while facilitating the EVCS in dispatching the load.

Future research can broaden the scope of this charging system in several directions. First, studies could investigate more comprehensive data sources, such as user travel patterns or regional energy policies, to better reflect real-world complexities. Second, the long-term impact of varying charging behaviors—such as user acceptance of dynamic pricing or shifts in mobility patterns—could be examined to assess the sustainability and scalability of the system. Third, incorporating social and economic factors could provide deeper insights into how charging decisions influence, and are influenced by, broader community interests. Finally, future research could explore the LLM itself further by addressing practical issues—such as investigating how the scale of an LLM influences both its performance and the resources required for implementation—to better bridge the gap between research prototypes and commercially viable solutions.

APPENDIX

GPT-4o-mini assumes that the user's probability of choosing a charging station is based on several factors: distance, rating, electricity price, charging speed, and availability of free charging slots. We can quantify these factors with scores and calculate the probability of selecting each station using a weighted average approach. The analysis steps are:

1) Distance: Closer stations have higher weight.
2) Electricity Price: Lower subsidized electricity prices are more appealing.
3) Preference Satisfied: EVCS with Charging mode satisfying the user's preference are preferred.
4) Availability of Free Slots: Stations with more available slots are easier to choose.

The evaluation model GPT-4o-mini used is shown in Table VII.

The probability for each EVCS is calculated by:

$$Prb_i = \frac{S_i}{\sum S_n} \tag{A1}$$

TABLE VII

EVALUATION MODEL PROVIDED BY GPT-4O-MINI

Factors	Weight	Teld	Xiaoyi	Yiwei
Price	40	35	38	27
Preference Satisfied	30	27	25	22
Distance	20	15	10	18
Availability	10	8	5	10
Total Score	100	85	78	77

TABLE VIII

EXAMPLE OF HOLIDAY'S IMPACT ON USER'S PREFERENCES

Preferences	Content	Value	Proportion (Workday)	Proportion (Holiday)
Start Charging Time	Morning	6:01-12:00	20%	25%
	Noon	12:01-18:00	20%	15%
	Evening	18:00-24:00	20%	25%
	Midnight	0:01-6:00	20%	25%
	Any	/	20%	10%
Charging Mode	Fast	/	70%	50%
	Slow	/	10%	30%
	Any	/	20%	20%
Charging Price	Low	[0.8,1.0]*r_t	50%	60%
	High	[1.0,1.2]*r_t	20%	15%
	Any	/	30%	25%
Start SOC	Low	[5,15]	30%	25%
	High	[16,30]	30%	25%
	Any	/	40%	50%

where S_i is the total score of EVCS$_i$, and n represents the total number of EVCS.

The influence of holidays and weather conditions on the user's preferences and how the User Agent would evaluate the charging decision is studied. An example of holiday's impact on user's preferences is provided in Table VIII and an example of weather conditions' impact on evaluation model is provided by GPT-4o-mini is provided in Table IX.

Holidays influence the user's preferences. In our work, we need proportion to simulate the real-world situation. However, if this system is implemented, real-world data can be directly taken into consideration and there is no need to manually make a distinction between workdays and holidays.

The difference in evaluation when facing various weather conditions is obvious and understandable. Similar to humans when making decisions, as weather conditions become less favorable, more attention is paid to distance and availability rather than price or preference satisfaction.

The loss of CGAN during the training process is shown in Figure 19.

TABLE IX

EXAMPLE OF WEATHER CONDITIONS' IMPACT ON EVALUATION MODEL PROVIDED BY GPT-4O-MINI

Factors	Weight (Sunny Day)	Weight (Cloudy Day)	Weight (Rainy Day)
Price	40	35	30
Preference Satisfied	30	30	25
Distance	20	25	30
Availability	10	10	15
Total Score	100	100	100

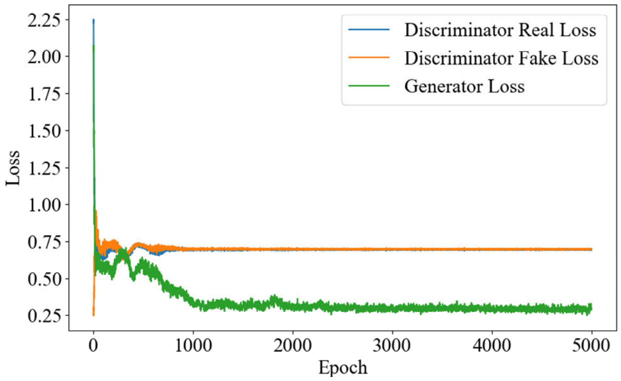


Fig. 19. Training loss of CGAN.

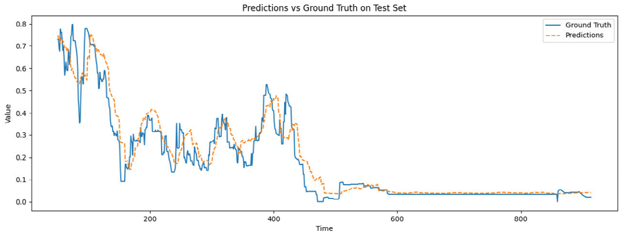


Fig. 20. Time-LLM's prediction results on test dataset.

The Time-LLM is trained based on private dataset and its results on test dataset are shown in Figure 20.

The results illustrate the Time-LLM's ability in temporal prediction, combining with text information. However, the results also performs certain level of lagging effect, indicating the model's dependency on historical data.

The overall score of methods is calculated by:

$$s_i = \frac{(a_1^{\min}/a_1^i) + (a_2^i/a_2^{\max}) + (a_3^i/a_3^{\max})}{3} \cdot 100 \qquad \text{(A2)}$$

where a_1^i, a_2^i, and a_3^i represent Average Charging Price, EVCS Revenue and Average Load Aligning Rate of method i respectively.

REFERENCES

[1] H. M. Salih et al., "A electric vehicle blockchain: Problems and opportunities," *Babylonian J. Netw.*, vol. 2024, pp. 18–24, Feb. 2024.

[2] S. Powell, G. V. Cezar, L. Min, I. M. L. Azevedo, and R. Rajagopal, "Charging infrastructure access and operation to reduce the grid impacts of deep electric vehicle adoption," *Nature Energy*, vol. 7, no. 10, pp. 932–945, Sep. 2022.

[3] H. I. Shaheen, G. I. Rashed, B. Yang, and J. Yang, "Optimal electric vehicle charging and discharging scheduling using metaheuristic algorithms: V2G approach for cost reduction and grid support," *J. Energy Storage*, vol. 90, Jun. 2024, Art. no. 111816.

[4] T. Mongaillard et al., "Large language models for power scheduling: A user-centric approach," in *Proc. 22nd Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, Jun. 2024, pp. 321–328.

[5] J. Feng, C. Cui, C. Zhang, and Z. Fan, "Large language model based agent framework for electric vehicle charging behavior simulation," 2024, *arXiv:2408.05233*.

[6] Z. Yan and Y. Xu, "Real-time optimal power flow with linguistic stipulations: Integrating GPT-agent and deep reinforcement learning," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 4747–4750, Mar. 2024.

[7] H. Qu et al., "ChatEV: Predicting electric vehicle charging demand as natural language processing," *Transp. Res. D, Transp. Environ.*, vol. 136, Nov. 2024, Art. no. 104470.

[8] X. Chen, X. Lu, Q. Li, D. Li, and F. Zhu, "Integration of LLM and human–AI coordination for power dispatching with connected electric vehicles under SAGVNs," *IEEE Trans. Veh. Technol.*, vol. 74, no. 2, pp. 1992–2002, Feb. 2025.

[9] M. Jin, B. Sel, F. Hardeep, and W. Yin, "Democratizing energy management with LLM-assisted optimization autoformalism," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Sep. 2024, pp. 258–263.

[10] Y. Talebirad and A. Nadiri, "Multi-agent collaboration: Harnessing the power of intelligent LLM agents," 2023, *arXiv:2306.03314*.

[11] Q. Liu, J. Mu, D. Chen, R. Zhang, Y. Liu, and T. Hong, "LLM enhanced reconfigurable intelligent surface for energy-efficient and reliable 6G IoV," *IEEE Trans. Veh. Technol.*, vol. 74, no. 2, pp. 1830–1838, Feb. 2025.

[12] Z. Lan, L. Liu, B. Fan, Y. Lv, Y. Ren, and Z. Cui, "Traj-LLM: A new exploration for empowering trajectory prediction with pre-trained large language models," *IEEE Trans. Intell. Vehicles*, early access, Jun. 2024, doi: 10.1109/TIV.2024.3418522.

[13] Y. Tian et al., "VistaGPT: Generative parallel transformers for vehicles with intelligent systems for transport automation," *IEEE Trans. Intell. Veh.*, vol. 8, no. 9, pp. 4198–4207, Sep. 2023.

[14] T. Cai et al., "Driving with regulation: Interpretable decision-making for autonomous vehicles with retrieval-augmented reasoning via LLM," 2024, *arXiv:2410.04759*.

[15] X. Zhou et al., "ElecBench: A power dispatch evaluation benchmark for large language models," 2024, *arXiv:2407.05365*.

[16] W.-C. Kang et al., "Do LLMs understand user preferences? Evaluating LLMs on user rating prediction," 2023, *arXiv:2305.06474*.

[17] L. Podo, M. Ishmal, and M. Angelini, "Vi(E)va LLM! A conceptual stack for evaluating and interpreting generative AI-based visualizations," 2024, *arXiv:2402.02167*.

[18] Y. Wang et al., "LLM-GAN: Construct generative adversarial network through large language models for explainable fake news detection," 2024, *arXiv:2409.01787*.

[19] X. Li, C. Zhao, H. Zhao, L. Wu, and M. HE, "GANPrompt: Enhancing robustness in LLM-based recommendations with GAN-enhanced diversity prompts," 2024, *arXiv:2408.09671*.

[20] D. Devadiga et al., "GLEAM: GAN and LLM for evasive adversarial malware," in *Proc. 14th Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2023, pp. 53–58.

[21] G. Dhingra, S. Sood, Z. M. Wase, A. Bahga, and V. K. Madisetti, "Protecting LLMs against privacy attacks while preserving utility," *J. Inf. Secur.*, vol. 15, no. 4, pp. 448–473, 2024.

[22] G. Graber, V. Galdi, V. Calderaro, F. Lamberti, and A. Piccolo, "Centralized scheduling approach to manage smart charging of electric vehicles in smart cities," in *Proc. 7th Int. Conf. Smart Cities Green ICT Syst.*, 2018, pp. 238–245.

[23] Q. Kang, J. Wang, M. Zhou, and A. C. Ammari, "Centralized charging strategy and scheduling algorithm for electric vehicles under a battery swapping scenario," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 659–669, Mar. 2016.

[24] M. Jin et al., "Time-LLM: Time series forecasting by reprogramming large language models," 2023, *arXiv:2310.01728*.

[25] Y. Cao, N. Wang, G. Kamel, and Y.-J. Kim, "An electric vehicle charging management scheme based on publish/subscribe communication framework," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1822–1835, Sep. 2017.

[26] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 9459–9474.

[27] Z. Jiang et al., "Active retrieval augmented generation," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, Jan. 2023, pp. 7969–7992.

[28] J. E. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, Jan. 2021, p. 3.

[29] D. Carraro and D. Bridge, "Enhancing recommendation diversity by reranking with large language models," *ACM Trans. Recommender Syst.*, Oct. 2024, doi: 10.1145/3700604.

[30] J. Liu, Z. Zhao, X. Luo, P. Li, G. Min, and H. Li, "SlaugFL: Efficient edge federated learning with selective GAN-based data augmentation," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 11191–11208, Dec. 2024.

[31] G. Sun et al., "Generative AI for advanced UAV networking," *IEEE Netw.*, early acess, Nov. 2024, doi: 10.1109/MNET.2024.3494862.

[32] S. Wang, S. Bi, and Y. J. A. Zhang, "Reinforcement learning for real-time pricing and scheduling control in EV charging stations," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 849–859, Feb. 2019.

Zeyuan Niu (Graduate Student Member, IEEE) received the B.Eng. degree in electrical engineering from the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2023. He is currently pursuing the M.S. degree with the School of Electrical Engineering, Shanghai Jiao Tong University. His research interests include smart grids, road-power coupling networks, multi-agent systems, generative AI models, and their applications in renewable energy dispatch and power system optimization.

Jiamei Li received the B.S. degree in electrical engineering from Sichuan University, Chengdu, China, in 2018, and the Ph.D. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2023. She is currently a Research Assistant with the School of Electrical Engineering, Shanghai Jiao Tong University. Her research interests include artificial intelligence in power systems, electricity markets, and virtual power plant.

Qian Ai (Senior Member, IEEE) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1991, the M.S. degree from Wuhan University, Wuhan, China, in 1994, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1999, all in electrical engineering. He was with Nanyang Technological University, Singapore, for one year; the University of Bath, Bath, U.K., for two years; and then with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, where he is currently a Professor. His current research interests include power quality, load modeling, smart grids, microgrids, and intelligent algorithms.

Jiamei Jiang, photograph and biography not available at the time of publication.

Qiunan Yang, photograph and biography not available at the time of publication.

Haojie Zhou, photograph and biography not available at the time of publication.