2026-01-0461 Published 07 Apr 2026

From LLM to Deep Learning: Efficient Simulation of Last-Mile Energy Behavior in Campus Communities

Zhifeng Yang, Yongjian Chen, and Shiqi(Shawn) Ou South China University of Technology

Citation: Yang, Z., Chen, Y., and Ou, S.S "From LLM to Deep Learning: Efficient Simulation of Last-Mile Energy Behavior in Campus Communities," SAE Technical Paper 2026-01-0461, 2026, doi:10.4271/2026-01-0461.

Received: 19 Oct 2025 Revised: 18 Jan 2026 Accepted: 27 Jan 2026

Abstract

Communities are critical nodes in the urban energy network, integrating energy, transportation, and building systems to enhance overall energy efficiency. Accurate management of these systems depends on characterizing individual energy-demand behaviors. However, traditional last-mile behavioral models often fail to capture the complex and non-linear nature of human decision-making, underscoring the need for advanced simulation techniques. Although agent-based simulations powered by Large Language Models (LLMs) have demonstrated considerable efficacy across diverse domains, their substantial computational demands, specifically in terms of token consumption, pose pronounced constraints in large-scale simulations involving tens of thousands of agents, considerably limiting their practical applicability. To address this challenge, a deep learning-based approach is proposed for single-step prediction of last-mile behavior. A hybrid architecture, consisting of an MLP encoder and a Transformer decoder, is constructed to fit behavioral data generated by LLM agents, thereby replacing the resource-intensive LLM simulation process. Additionally, a hierarchical weighted loss function and a causal masking mechanism are designed to optimize training, and Bayesian optimization is introduced for automated hyperparameter tuning. Experimental results indicate that the proposed method achieves an average accuracy of 0.8881 across multiple hierarchical behavior predictions and an overall accuracy of 0.7576, significantly outperforming traditional long-sequence prediction methods. Moreover, model performance is further enhanced through Bayesian optimization. The proposed framework provides an efficient and scalable solution for large-scale energy behavior simulation, demonstrating strong practical value and promising broader applicability.

Introduction

Against the backdrop of the deepening global energy transition, optimizing energy structures and improving energy efficiency have become central priorities. To achieve this goal, enhancing energy management, particularly achieving granular energy management, is a critical pathway for boosting energy utilization efficiency and promoting low-carbon development. As key hubs of urban end-use energy consumption, communities exhibit a deeply coupled nature among their energy, building, and transportation systems. This makes them critical nodes in the urban energy network and plays a significant role in realizing holistic urban energy optimization management. In community energy management, accurately characterizing individual energy-demand behaviors is especially crucial, and constructing corresponding behavioral models provides an effective means to achieve this. By modeling individual energy consumption behaviors at the last mile, more precise and efficient energy dispatch can be supported based on specific behavioral characteristics.

Currently, the Agent-based Model (ABM) has garnered significant attention as a common method for simulating energy consumption behaviors. As a complex system modeling approach, ABM comprises multiple interacting and autonomous entities, enabling effective simulation of energy use behaviors among various agent modules within energy systems. For instance, Chen et al. [1] developed an ABM-based simulation model to analyze heat pump usage behaviors in residential communities; Chingcuanco et al. [2] constructed an urban energy consumption model using ABM to simulate short- and long-term energy decisions of enterprises and households; Tian et al. [3] integrated deep learning with ABM for household energy consumption simulation; Ding et al. [4] employed ABM to simulate energy consumption behaviors in university dormitories; Qiao et al. [5] simulated energy use patterns of students in campus study rooms; and Zhang et al. [6] used ABM to generate energy demand scenarios for community residents, simulating energy consumption activities across different types of apartments and households. Although ABM has achieved

certain success in the studies mentioned above, it typically relies on a rule-driven approach that guides agent behaviors through predefined rules. This makes it suitable for simple, deterministic scenarios but inadequate for comprehensively and accurately capturing the complexity and dynamic nature of real-world energy consumption behaviors.

In recent years, the rise of Large Language Models (LLMs) has offered new perspectives for modeling energy consumption behaviors. LLMs possess powerful language comprehension and generation capabilities, enabling them to process complex textual information and uncover underlying patterns and correlations [7]. Compared to rule-driven approaches, LLMs are better equipped to handle uncertainty and ambiguity in data [8], dynamically adapt to complex and variable energy use scenarios, and thereby simulate energy consumption behavior patterns of both individuals and groups more accurately. Although LLM-based agent modeling methods have not yet been reported in the energy domain, they have demonstrated considerable feasibility in behavioral modeling and simulation studies in other fields.

In social science research, studies have focused on endowing LLMs with specific personality traits, enabling them to generate behavioral data as independent entities [9, 10]. Consumer behavior research indicates that LLM agents align with economic theory expectations across multiple dimensions, including downward-sloping demand curves, diminishing marginal returns to income, and state dependence [11]. Argyle et al. [12] demonstrated that GPT-3-based agents can accurately simulate human responses from different social subgroups with complex thoughts, attitudes, and socio-cultural backgrounds. Gao et al. [13] utilized ChatGLM-driven LLM agents to simulate information diffusion in social networks. Their experimental results showed the LLM agents' ability to replicate empirical behaviors observed in real networks, highlighting the potential of LLM in social interaction simulation.

In game theory research, LLM agents have also been introduced into classical game experiments [14] to explore competitive and cooperative behaviors. Aher et al. [15] constructed multiple agents based on different LLMs to simulate classic experiments in economics, psycholinguistics, and social psychology. They found that in the ultimatum game, the decision-making tendencies of LLM agents closely align with those of humans. Brookins et al. [16] observed in the dictator game that LLM agents generally exhibited fairer allocation behavior than humans in most cases; meanwhile, in one-shot prisoner's dilemma experiments [16], the average cooperation rate of LLM agents reached 65.4%, significantly higher than the 37% observed among human participants.

Despite the demonstrated effectiveness of LLM agents across various domains, their application in large-scale simulations still faces the challenge of immense computational resource consumption, particularly in terms of token usage. While some existing studies reduce the simulation scale to alleviate computational burden, many practically significant "emergent behaviors" precisely rely on interaction simulations within large-scale populations. Therefore, maintaining the authenticity of population-scale simulations is crucial. Deep learning models can learn complex mapping relationships through training and achieve accurate predictions. Suppose a deep learning model can be constructed based on the "last-mile" behavioral data generated by LLM agents and deployed for rapid inference in subsequent large-scale simulations. In that case, it holds promise for significantly reducing token consumption. However, no relevant research has been reported in this direction to date.

To address this gap, this paper proposes a deep learning-based method for generating agent last-mile energy consumption behaviors, trained using simulation data from LLM agents. The core contributions are summarized as follows:

1. **An efficient deep learning-based method for last-mile behavior generation is proposed.** A hybrid architecture, consisting of an MLP feature encoder and a Transformer-based hierarchical decoder, is constructed to fit the last-mile behavioral data generated by LLM agents, thereby replacing the resource-intensive LLM simulation process.
2. **A hierarchical weighted loss function and a causal masking mechanism are designed.** The former addresses hierarchical dependencies in behavioral decisions, while the latter mitigates information leakage during sequence generation. Together, these enhancements improve the model's prediction accuracy across key hierarchies, such as building categories and equipment actions.
3. **Bayesian optimization is introduced for automated hyperparameter tuning.** This approach significantly enhances the model's overall and average accuracy on the validation set, validating the effectiveness of the proposed architecture in modeling complex behavioral sequences.

LLM Agents Dataset

Common types of communities primarily include industrial parks, campus communities, and residential communities. Among these, campus communities are well-suited for modeling energy consumption behavior due to their diverse population types, relatively regular energy usage patterns, and well-established energy infrastructure with clearly defined functional zones. Therefore, this study selects a campus community as the research subject and constructs a simulation environment based on the Guangzhou International Campus of South China University of Technology. Within this environment, the attributes of representative LLM agents are defined based on the actual roles and behavioral characteristics of various individuals on the campus. Using an LLM-driven

approach, a simulation dataset is constructed to support the generation of last-mile energy consumption behavior, facilitating the training and validation of subsequent deep learning models.

LLM Agents Input and Output

To simulate human decision-making and logical thinking and drive LLMs to exhibit reasonable energy consumption behaviors, the input to LLM agents includes the current time, agent Settings, and the agent's daily plan, etc., as shown in Equations (1) to (4).

$$\mathrm{LLM}_{\mathrm{input}} = \left(\mathrm{Time}, \mathrm{Agent}_{\mathrm{info}}, \mathrm{Agent}_{\mathrm{schedule}}\right) \tag{1}$$

$$\mathrm{Agent}_{\mathrm{info}} = (\mathrm{Role}, \mathrm{Gender}, \mathrm{Age}, \mathrm{School}, \mathrm{Major}, \mathrm{Address}, \mathrm{Workplace}, \mathrm{Personality})\# \tag{2}$$

$$\mathrm{Personality} = (\mathrm{Work}_{\mathrm{intensity}}, \mathrm{Social}_{\mathrm{preference}}, \mathrm{Thinking}_{\mathrm{style}}, \mathrm{Planning}_{\mathrm{style}}) \tag{3}$$

$$\begin{cases} \mathit{Work}_{\mathrm{intensity}} \in \{\mathrm{High}, \mathrm{Low}\} \\ \mathit{Social}_{\mathrm{preference}} \in \{\mathrm{Extroverted}, \mathrm{Introverted}\} \\ \mathit{Thinking}_{\mathrm{style}} \in \{\mathrm{Rational}, \mathrm{Emotional}\} \\ \mathit{Planning}_{\mathrm{style}} \in \{\mathrm{Organized}, \mathrm{Spontaneous}\} \end{cases} \tag{4}$$

where, ***Time*** represents the current timestamp in the format "2025-08-24 Monday 00:00:00", which is encoded as a seven-dimensional feature vector comprising month, day, day of the week, hour, minute, whether it is a weekend, and time period (late night/morning/afternoon/evening).

Agent*$_{\mathbf{info}}$** is a vector describing specific characteristics of different agents. ***Role encompasses nine population types: undergraduate students, graduate students, teachers, administrative staff, cafeteria staff, convenience store staff, school hospital staff, security staff, and visitors. ***Age*** is assigned within realistic age ranges for each role. ***School*** and ***Major*** are attributes only assigned to undergraduate and graduate students. ***Address*** is included only for agents with on-campus dormitory assignments, while ***Workplace*** is specified only for faculty and staff members. ***Personality*** is a four-dimensional vector that further differentiates among agents, including attributes such as $\mathit{Work}_{\mathrm{intensity}}$, $\mathit{Social}_{\mathrm{preference}}$, $\mathit{Thinking}_{\mathrm{style}}$, and $\mathit{Planning}_{\mathrm{style}}$. These characteristics may influence their energy consumption behavior patterns. Thus, the $\mathit{Agent}_{\mathrm{info}}$ vector has a total of 11 dimensions.

***Agent*$_{\mathbf{schedule}}$** represents the agent's planned activity per 30-minute interval, formulated as a one-dimensional feature with possible values including: attending class, conducting research, performing experiments, working, studying, resting, entertainment, exercising, breakfast, lunch, dinner, parking, and electric vehicle charging. Collectively, *Time*, $\mathit{Agent}_{\mathrm{info}}$, and $\mathit{Agent}_{\mathrm{schedule}}$ constitute the $\mathit{LLM}_{\mathrm{input}}$, resulting in a combined vector dimensionality of 19.

The LLM agents' decision outputs must strictly adhere to the hierarchical structure and constraints defined by the predefined knowledge graph. The decision-making process follows a top-down, hierarchical sequence: First, select the $\mathit{Building}_{\mathrm{category}}$; Then determine the specific *Building*, *Floor*, and *Room*; Finally, select a particular piece of *Equipment* within the *Room* and determine its operational *Action*. This complete decision sequence collectively forms the structured output vector llm_output, as shown in Equations (5) to (8).

$$\mathrm{LLM}_{\mathrm{output}} = \begin{pmatrix} \mathrm{Building}_{\mathrm{category}}, \mathrm{Building}, \mathrm{Floor}, \mathrm{Room}, \\ \mathrm{Equipment}, \mathrm{Action} \end{pmatrix} \tag{5}$$

$$\mathrm{Action} = \left(\mathrm{Equipment}_{\mathrm{action}}, \mathrm{Pre\_Equipment}_{\mathrm{action}}\right) \tag{6}$$

$$\mathit{Equipment}_{\mathrm{action}} \in \left\{\mathrm{Turn}_{\mathrm{on}}, \mathrm{Turn}_{\mathrm{off}}, \mathrm{No}_{\mathrm{action}}\right\} \tag{7}$$

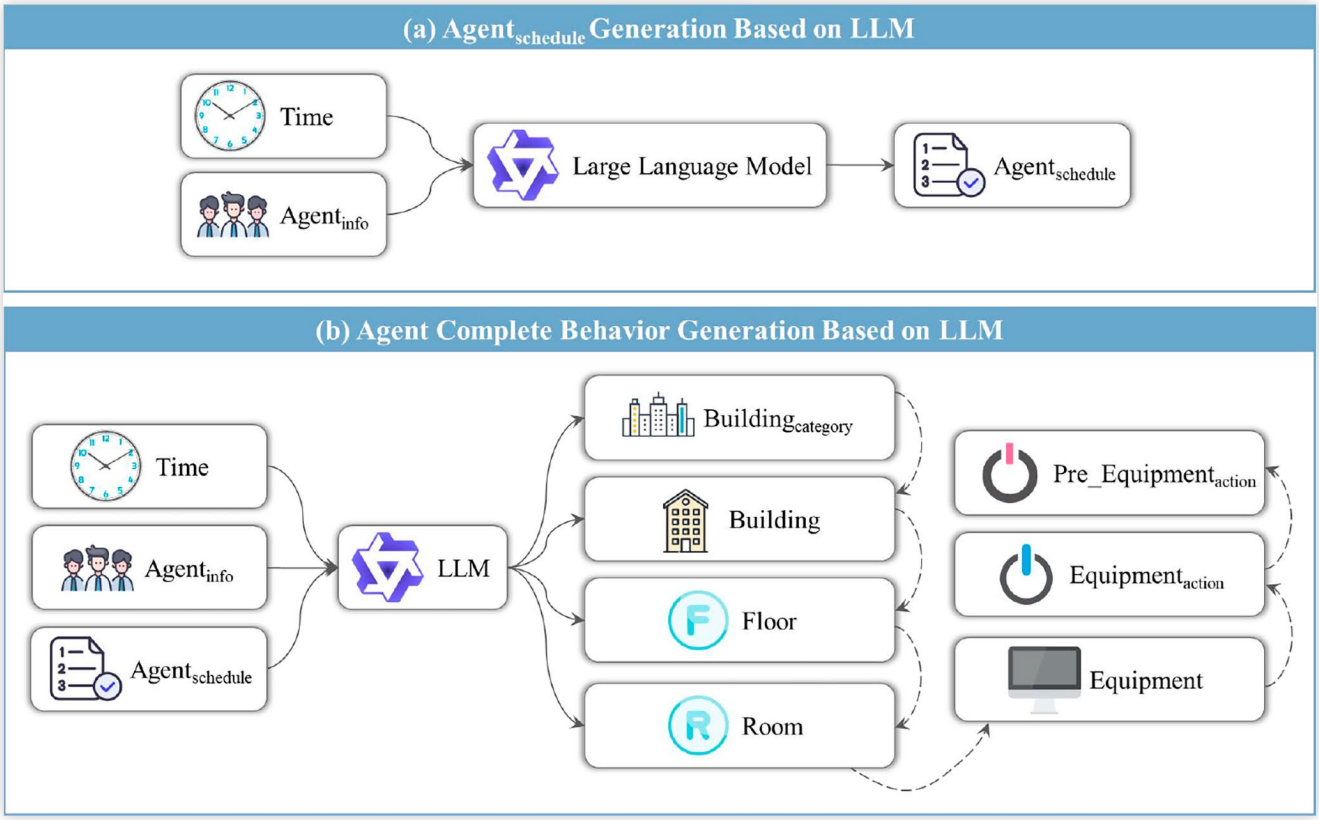
$$\mathrm{Pre\_}\mathit{Equipment}_{\mathrm{action}} \in \{\mathrm{True}, \mathrm{False}\} \tag{8}$$

where, $\mathit{Building}_{\mathrm{category}}$, *Building*, *Floor*, *Room*, and *Equipment*, strictly conforms to the structure of the knowledge graph, which requires predefining the floors, rooms, and equipment within different buildings. $\mathit{Equipment}_{\mathrm{action}}$ represents the operation performed on the current equipment, with the decision space including $\mathrm{Turn}_{\mathrm{on}}$, $\mathrm{Turn}_{\mathrm{off}}$, and $\mathrm{No}_{\mathrm{action}}$. $\mathit{Pre\_Equipment}_{\mathrm{action}}$ indicates whether to turn off the previously activated equipment, with the decision space comprising True and False. Thus, $\mathit{LLM}_{\mathrm{output}}$ is a seven-dimensional vector.

Dataset Construction Pipeline

An agent set comprising 144 individuals across nine population roles was constructed, with the respective counts for each role as follows: 40 undergraduate students, 40 graduate students, 16 faculty members, 16 administrative staff, 8 cafeteria staff, 8 convenience store staff, 8 campus clinic staff, 4 security personnel, and 4 visitors. Values for each indicator in the *Personality* vector were randomly sampled from their respective domains. As illustrated in Figure 1(a), the $\mathit{Agent}_{\mathrm{schedule}}$ is determined through LLM decision-making based on inputs of *Time* and $\mathit{Agent}_{\mathrm{info}}$, thereby forming the complete $\mathit{LLM}_{\mathrm{input}}$. Subsequently, the $\mathit{LLM}_{\mathrm{input}}$ is used as a prompt fed into the LLM, which performs hierarchical decision-making based on the building structure, ultimately yielding an operational action for a specific piece of equipment. Finally, based on Qwen2.5:7b and the pipeline as shown in Figure 1, a last-mile behavior simulation of LLM agents was conducted with a step size of 30 minutes over a total duration of three days. After data cleaning, 18944 valid data records were obtained for subsequent deep learning model training.

FIGURE 1 LLM Agents Dataset Construction Pipeline



Deep Learning Architecture

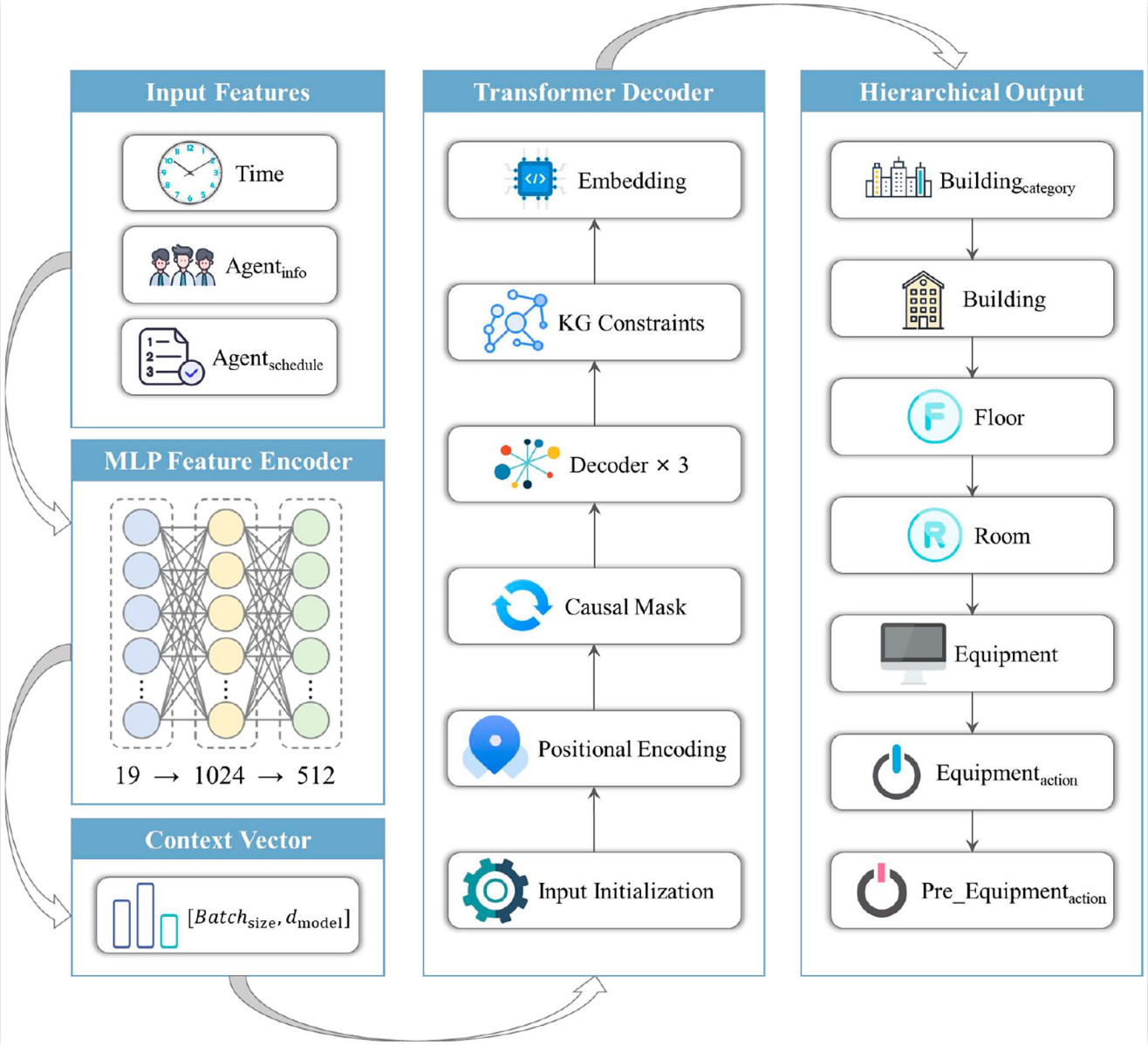
This paper aims to construct an efficient deep learning model (as shown in Figure 2) to replace computationally expensive LLMs in large-scale population simulations, enabling the rapid generation of individual energy-use behaviors. Given that the input to the LLM consists of structured features and the output comprises behavioral sequences with explicit hierarchical relationships, this study adopts a single-step prediction approach to directly generate complete hierarchical behavioral sequences from contextual information at the current moment. This strategy avoids the noise and redundancy introduced by historical dependencies in traditional sequence modeling. To this end, a hybrid architecture integrating an MLP feature encoder and a Transformer decoder is proposed to accommodate the heterogeneity of input features and effectively capture hierarchical dependencies in the output sequences. Specifically, the MLP encoder maps multi-dimensional input features into a unified high-dimensional contextual representation, while the Transformer decoder, leveraging a causal masking mechanism, autoregressively generates structured behavioral sequences. Furthermore, given the varying importance of hierarchical levels in behavioral decision-making, a hierarchically weighted loss function is constructed to improve the predictive accuracy of key levels, thereby enhancing overall modeling performance and semantic coherence.

MLP Feature Encoder

The Multilayer Perceptron (MLP) [17, 18], a foundational yet powerful deep learning architecture, is employed in this study as a feature encoder. Its role is to map heterogeneous and multi-source agent features into a unified high-dimensional representation, thereby providing structured contextual information to the subsequent Transformer decoder. The MLP encoder described in this paper uses a three-layer fully connected neural network, which effectively captures complex nonlinear relationships among input features. The MLP encoder maps the agent's heterogeneous features into a unified high-dimensional representation and serves as the core function to provide structured context information for the subsequent decoding process. It helps the model capture the complex patterns in behavioral decisions more effectively and improves the prediction efficiency.

Specifically, the first layer of the encoder maps the 19-dimensional input features to a 1024-dimensional hidden space, enhancing the model's expressive power through the introduction of a non-linear activation function. The second layer further compresses the feature dimension to 512, progressively extracting the most discriminative feature information. The final output layer maps the features to 256 dimensions, aligning with the input dimension of the subsequent Transformer decoder to ensure consistency in information transfer. The entire encoding process utilizes ReLU as the activation function and incorporates Dropout regularization (0.1) after each

FIGURE 2 Deep Learning Architecture



layer to mitigate overfitting and improve generalization. It is noteworthy that the encoder applies Layer Normalization (LayerNorm) after the output layer, which not only helps stabilize the training process but also standardizes feature distributions, providing more stable input conditions for the decoding stage.

The forward propagation process of the MLP encoder can be formally represented as follows. Let the input feature vector be $x \in R^{19}$, then the encoding process is shown in Equations (9) to (11).

$$h_1 = \text{Dropout}\left(\text{ReLU}\left(W_1 x + b_1\right)\right) \tag{9}$$

$$h_2 = \text{Dropout}\left(\text{ReLU}\left(W_2 h_1 + b_2\right)\right) \tag{10}$$

$$\text{context} = \text{LayerNorm}\left(W_3 h_2 + b_3\right) \tag{11}$$

where, h_1 denotes the first hidden state with a dimensionality of R^{1024}, representing the first non-linear transformation of the input features. W_1 is the weight matrix of the first layer with a dimensionality of $R^{1024 \times 19}$, which maps the input features to the first hidden space. b_1 is the bias vector of the first layer with a dimensionality of R^{1024}. The final output is the context vector *context* with a dimensionality of R^{256}, serving as a compact representation of the agent features.

The design of the MLP encoder fully accounts for the heterogeneity of input features and their semantic correlations in the behavior prediction task, laying a crucial foundation for an efficient, scalable last-mile behavior generation model.

Transformer Hierarchical Decoder

Transformer architecture was introduced by Vaswani et al. [19] in 2017, with its core innovation lying in a sequence modeling approach based entirely on the attention mechanism [20, 21]. The Transformer-based hierarchical decoder serves as the core component of this method, generating behavioral sequences with explicit hierarchical structures in an autoregressive manner from the contextual representation produced by the MLP encoder [22, 23]. This decoder retains the basic framework of the standard Transformer decoder. It is specifically optimized for the hierarchical characteristics of energy-use behavior decision-making, enabling effective modeling of the structured dependencies from building category selection to specific equipment operations.

In campus energy-use behavior simulations, the decision-making process of agents exhibits typical hierarchical constraints: higher-level choices (e.g., building category) constrain lower-level decisions (e.g., specific equipment operations). To capture such dependencies, the decoder employs a multi-head self-attention mechanism, allowing it to dynamically focus on the already generated portions of the sequence when generating each hierarchical level, thereby maintaining the semantic consistency of the behavior path. Specifically, when predicting each hierarchical level, the model captures relational information at different positions in the sequence in parallel through multiple attention heads. For instance, some attention heads may focus on identifying constraint relationships between building types and equipment, while others may be responsible for inferring the temporal or spatial plausibility of behaviors.

To prevent future information leakage during decoding, a causal masking mechanism is incorporated into the model. This means that during the autoregressive generation process, the model cannot access information from future positions when predicting the current position's output. For example, when predicting the *Building* level, it cannot access information from the *Floor* level. The causal mask achieves this constraint by setting the upper triangular part of the attention matrix to negative infinity, as illustrated in Equation (12).

$$\mathrm{mask}_{i,j} = \begin{cases} 0, \text{if } i \geq j \\ -\infty, \text{if } i < j \end{cases} \tag{12}$$

where, $mask_{i,j}$ represents the value at position (i, j) in the mask matrix. When $i \geq j$, $mask_{i,j}=0$, indicating no masking is applied, and the original attention weight is retained; when $i < j$, $mask_{i,j}=-\infty$, meaning full masking, as $-\infty$ becomes zero after the softmax calculation, completely preventing attention to future information. Consequently, the causal mask ensures that the model's generation process maintains temporal order, thereby preventing information leakage.

Through the design mentioned above, the Transformer-based hierarchical decoder can accurately model the complex hierarchical logic in behavioral decision-making while maintaining the autoregressive nature of the generation process. This provides reliable, structured reasoning capabilities for generating plausible behavioral sequences that conform to knowledge graph constraints.

Loss Function

In the last-mile energy behavior generation task, the model needs to simultaneously predict seven behavioral dimensions, each with strict semantic hierarchies. Different hierarchies not only exhibit structural dependencies but also vary in the impact of prediction errors. For instance, an error in predicting the Building category would invalidate all subsequent hierarchical decisions. In contrast, selecting different floors or rooms within the same building category shows a degree of interchangeability. Therefore, traditional uniformly weighted loss functions are inadequate for addressing the specific characteristics of hierarchical decision-making in this task.

To address this issue, this paper proposes a hierarchical weighted loss function that assigns differentiated weights to different hierarchies, enabling the model training to focus more on key decision levels that have a greater impact on the plausibility of the overall behavior. The loss function is defined as the weighted sum of individual hierarchical losses, as shown in Equation (13).

$$L_{\text{total}} = \sum_{l=1}^{L} w_l \cdot L_l \tag{13}$$

where, w_l represents the weight of the l-th level, and L_l represents the loss of the l-th level. For each level, the cross-entropy loss is adopted as the basic loss function:

$$L_l = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{V_l} y_{i,l,c} \log\left(\hat{y}_{i,l,c}\right) \tag{14}$$

where, N represents the batch size, V_l is the vocabulary size of each level l-th, $y_{i,l,c}$ represents the true label (one-hot encoding) of the i-th sample in the c-th category of the l-th level, and $\hat{y}_{i,l,c}$ represents the model's corresponding prediction probability.

Table 1 presents the weight coefficients assigned to different hierarchical levels. Specifically, the $Building_{\text{category}}$ level is particularly critical since its misprediction would lead to errors in all subsequent predictions; thus, it is assigned a weight of 1.0. In addition, since the decision results of *Equipment* and $Equipment_{\text{action}}$ will directly affect energy consumption and are the core steps of energy usage behavior; their weights are set to 1.0.

In contrast, for the same category of $Building_{\text{category}}$, different selections of *Building*, *Floor*, or *Room* remain reasonable as long as they belong to the same functional type. For instance, adjacent classrooms or laboratories on different floors reflect the interchangeability of specific spatial locations within the same functional category. Therefore, the weights for *Building*, *Floor*, and *Room* are set to 0.5. Although $Pre\_Equipment_{\text{action}}$ is important, it

TABLE 1 Weight Settings at Different Levels

Metric	$Building_{category}$	Building	Floor	Room
Weight	1.0	0.5	0.5	0.5
Metric	*Equipment*	$Equipment_{action}$	$Pre\_Equipment_{action}$	
Weight	1.0	1.0	0.5	

does not fundamentally impact the agent's core decision-making and is accordingly assigned a weight of 0.5.

Bayesian Optimization and Experimental Results

Bayesian Optimization

The deep learning model proposed in this paper features a complex architecture involving multiple critical hyperparameters. To fully exploit the model's potential and improve its fitting accuracy, this study incorporates Bayesian optimization [24, 25, 26] during model training to enable automated hyperparameter tuning. The core principle of Bayesian optimization is to formulate the hyperparameter tuning task as a black-box optimization problem. It intelligently guides the search process by constructing a probabilistic surrogate model of the objective function, thereby identifying superior hyperparameter configurations within a limited number of evaluations.

The optimization objective is as specified in Equation (15).

$$\theta^* = \arg\min_{\theta} E\left[L\left(y, f\left(x;\theta\right)\right)\right] \tag{15}$$

where, θ^* represents the hyperparameter combination to be optimized, and L denotes the loss function. Bayesian optimization effectively balances the trade-off between "exploration" (probing regions of high uncertainty in the parameter space) and "utilization" (concentrating the search near the currently best-performing parameter regions) by iteratively updating the posterior distribution over the objective function. Compared to computationally expensive grid search or less efficient random search, Bayesian optimization can identify superior hyperparameter configurations with fewer model evaluations, making it particularly suitable for deep learning models with high training costs.

In this study, the specific Bayesian optimization implementation uses the Tree-structured Parzen Estimator (TPE) as the acquisition function. The core strategy of TPE is to partition the hyperparameter space into "good-performing" (i.e., where the objective function value is below a certain threshold y^*) and "poor-performing" regions based on historical evaluation results. It then guides the selection of the next hyperparameters by comparing the conditional probability densities of these two regions. The calculation of its acquisition function $\alpha(\theta)$ is shown in Equation (16).

$$\alpha\left(\theta\right) = \frac{p\left(\theta \mid y < y^*\right)}{p\left(\theta \mid y \geq y^*\right)} \tag{16}$$

where, $\alpha(\theta)$ measures the value level of the hyperparameter configuration θ. A larger value of $\alpha(\theta)$ suggests that the hyperparameter configuration θ is more promising to explore. y^* represents the current optimal target value.

Another advantage of TPE is that it can handle different types of hyperparameters (continuous, discrete, and classified), which is particularly important for optimizing hyperparameters in deep learning models. In this task, the values of learning rate, dropout, and weight decay are uniformly distributed logarithmically. All other hyperparameters are taken from discrete sets. After Bayesian optimization, the final search results of the hyperparameters are shown in Table 2.

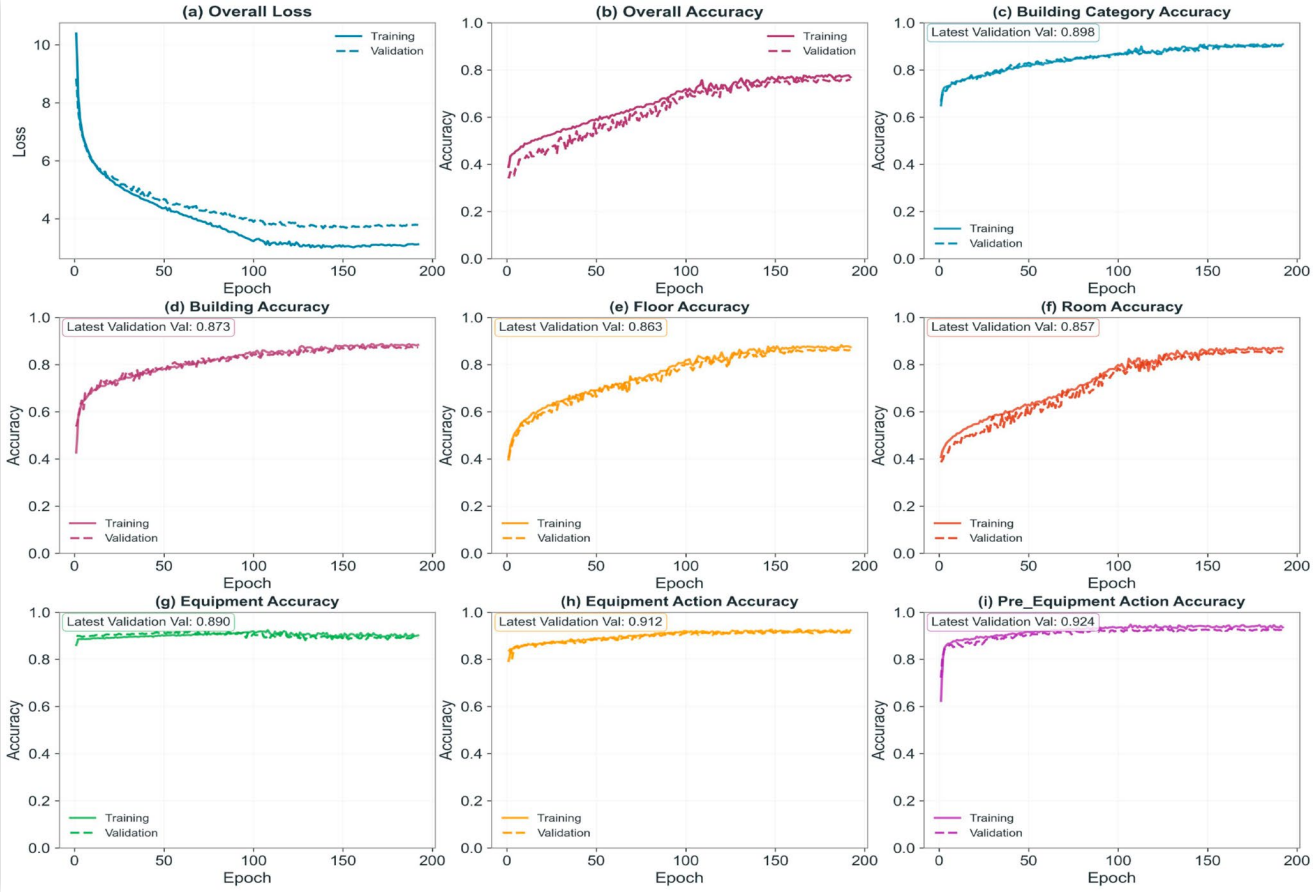
Model Accuracy Analysis

To evaluate the effectiveness of the proposed model, the constructed dataset was partitioned into training and validation sets at an 8:2 ratio. The model was trained for a maximum of 1000 epochs and included an early stopping strategy (patience of 50). Training was conducted using the optimal hyperparameter combination obtained via Bayesian optimization. The changes in accuracy and loss during the training process are shown in Figure 3. As illustrated, the training process remained stable and converged after 192 epochs, indicating favorable training efficiency and stability.

TABLE 2 Hyperparameter Values for Bayesian Optimization

Hyperparameter	Search Scope	Optimized Value
Learning rate	[0.00001, 0.01]	0.00066
Batch size	512, 1024, 2048, 4096	4096
Transformer Hidden Layer Dimensions	256, 512, 768, 1024	256
Num heads	4, 8, 12, 16	12
Encoder hidden dims 0	512, 1024, 1536, 2024	512
Encoder hidden dims 1	256, 512, 768, 1024	1024
Decoder num layers	2, 3, 4, 5, 6	5
Feedforward network dimension	1024, 2048, 3072, 4096	1024
Dropout	[0.05, 0.3]	0.07212
Weight decay	[0.000001, 0.001]	3.87211

FIGURE 3 Accuracy and Loss Value Change During Training



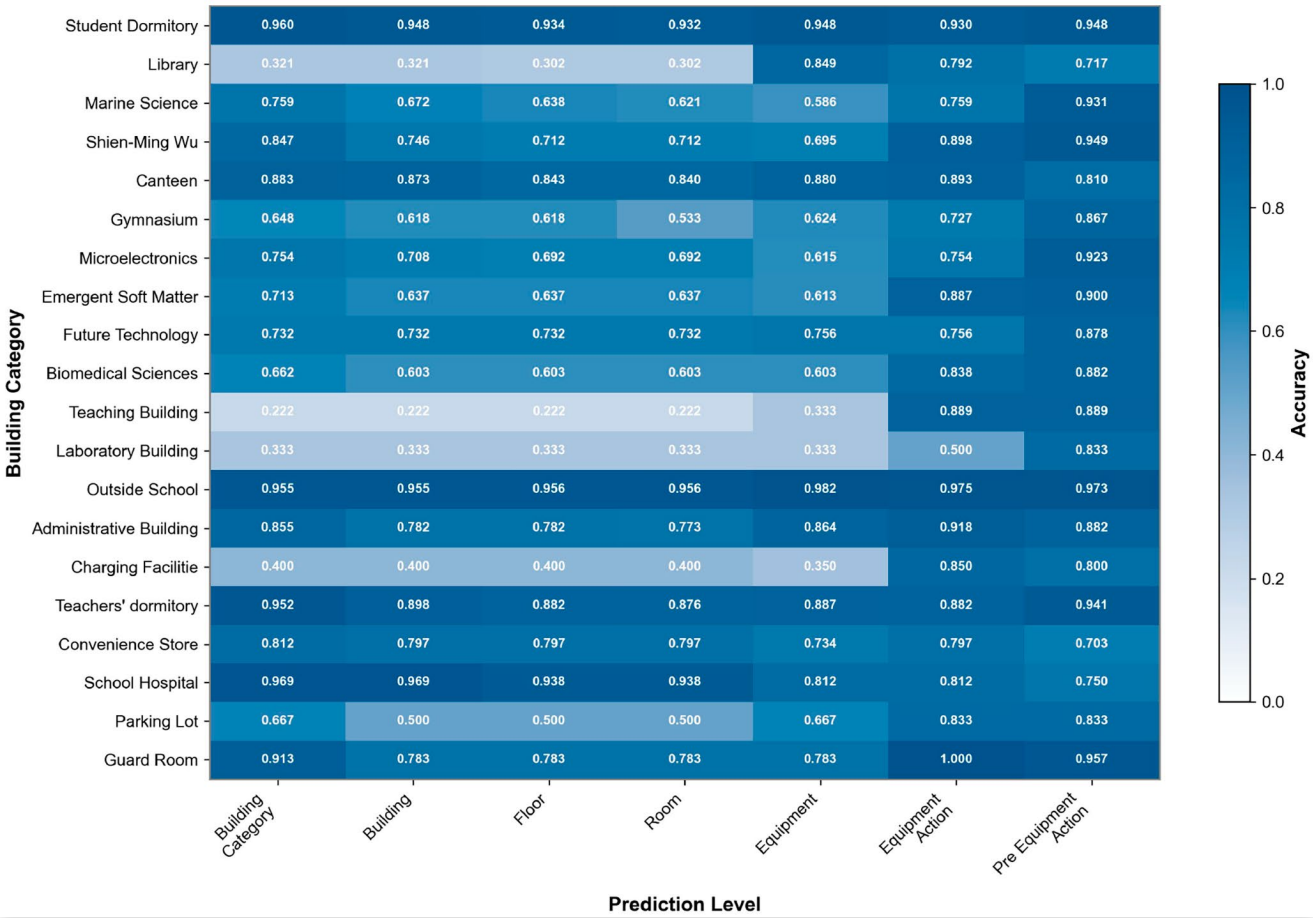
To comprehensively evaluate model performance, this study tests the trained models on the validation set and calculates prediction accuracy for each hierarchical behavior, summarizing the results in Table 3. Here, "Overall" denotes the overall correctness rate, the proportion of samples in which all seven hierarchical predictions are correct; "Average" refers to the average accuracy, calculated as the arithmetic mean across the seven hierarchies.

As shown in Table 3, the baseline "MLP Encoder + Transformer Decoder" architecture proposed in this paper has already achieved strong performance, with average and overall accuracies of 0.8539 and 0.6929, respectively. After incorporating Bayesian optimization, the model's performance further improved: the average accuracy increased to 0.8881, the overall accuracy rose to 0.7576, and the prediction accuracy across all seven hierarchical levels improved markedly. This confirms that Bayesian optimization successfully identified a superior set of hyperparameters, significantly boosting the model's fitting capability. It is particularly noteworthy that the accuracy for key hierarchical levels, such as $Equipment_{action}$ and $Pre\_Equipment_{action}$ both exceeded 0.9, and the accuracy for all levels was above 0.85, demonstrating the model's exceptional performance in complex hierarchical decision-making tasks.

TABLE 3 Comparison of The Accuracy of Different Methods

Metric/Accuracy Rate	MLP Encoder + Transformer Decoder	MLP Encoder + Transformer Decoder (Bayesian)	Transformer Encoder + Linear Regression
Average	0.8539	**0.8881**	0.3921
Overall	0.6929	**0.7576**	0.1159
$Building_{category}$	0.8701	**0.8978**	0.5215
Building	0.8357	**0.8727**	0.3044
Floor	0.8221	**0.8629**	0.2008
Room	0.8128	**0.8565**	0.1894
Equipment	0.8462	**0.8904**	0.3805
$Equipment_{action}$	0.8837	**0.9119**	0.7627
$Pre\_Equipment_{action}$	0.9068	**0.9243**	0.5536

FIGURE 4 Heat Maps of Prediction Accuracy at Different Levels



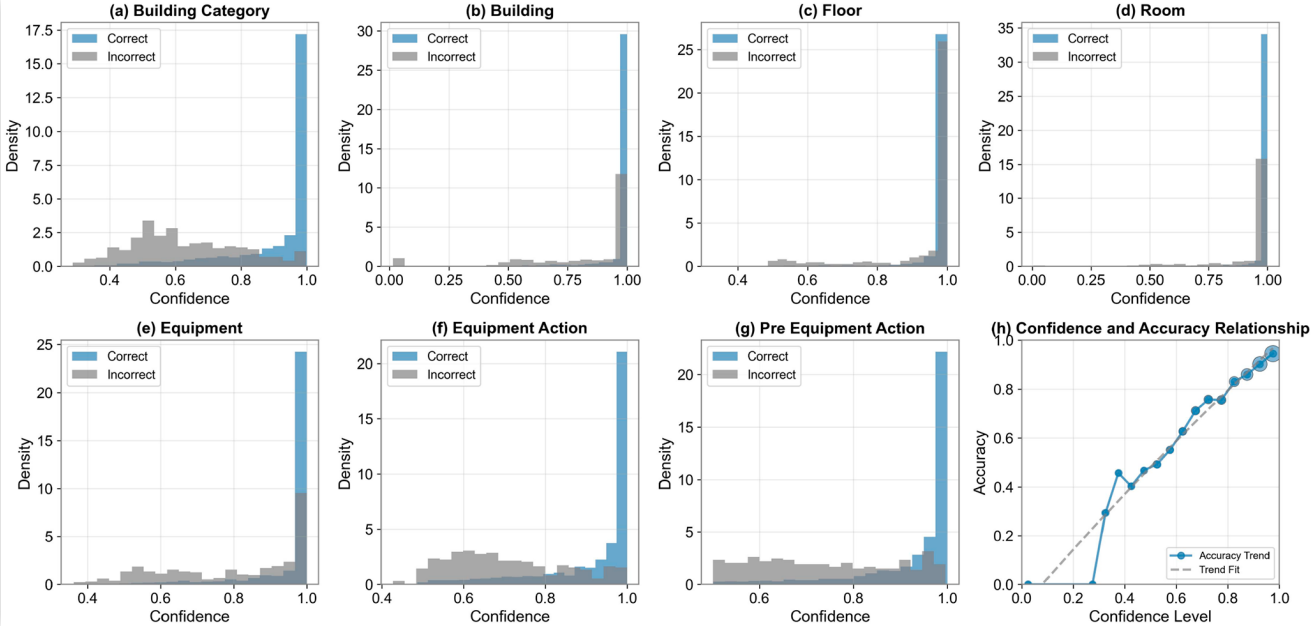
For comparison, we also evaluated a traditional long-sequence prediction method-the "Transformer Encoder + Linear Regression" model. This method aims to capture dependencies in historical sequences using a Transformer encoder, with an input sequence length of 10 for this task. However, as shown in Table 3, the prediction accuracy of this method was substantially lower than that of our proposed approach, with an average accuracy of only 0.3921 and an overall accuracy of 0.1159. This significant performance gap validates the decision to frame the problem as a single-step prediction task in this study. The underlying reason is that the model input in this study already includes $Agent_{\text{schedule}}$, and the behaviors simulated by the LLM agents are highly consistent with $Agent_{\text{schedule}}$. Therefore, the key to behavior prediction lies in modeling the current contextual information (particularly $Agent_{\text{schedule}}$), rather than the historical behavior sequence. In some cases, historical sequence patterns, which are inconsistent with $Agent_{\text{schedule}}$, may even introduce noise and adversely affect prediction. The experimental results robustly demonstrate that, within the context of this task, the adopted single-step prediction paradigm holds a distinct advantage over long-sequence prediction methods.

Furthermore, Figure 4 presents the distribution of prediction accuracy across building categories and hierarchies in a heatmap. Overall, most building categories and their corresponding sub-hierarchies achieve high prediction accuracy. Specifically, for the $Building_{\text{category}}$ hierarchy, 16 out of 20 categories achieve accuracy above 0.6; for the *Building*, *Floor*, and *Room* hierarchies, 15 categories each exceed 0.6 accuracy; for the *Equipment* and $Equipment_{\text{action}}$ hierarchies, 17 and 19 categories, respectively, surpass 0.6 accuracy. The $Pre\_Equipment_{\text{action}}$ hierarchy demonstrates the most robust performance, with all categories achieving accuracy above 0.7. These results further confirm the model's strong generalization capability across diverse scenarios and hierarchies.

Model Confidence Analysis

The model's overall accuracy has reached a satisfactory level, as evidenced by its predictive performance across various hierarchical levels, and it demonstrates particularly strong performance at critical decision levels (such as $Equipment_{\text{action}}$). To further analyze the model's decision-making reliability, Figure 5 illustrates the confidence distributions of prediction outcomes across various hierarchies. Overall, the model generally exhibits high confidence for correctly predicted samples, with their distribution distinctly right-skewed and concentrated in the 0.8–1.0

FIGURE 5 Comparison of Prediction Confidence Levels at Different Levels



interval. In contrast, for incorrect predictions, the model's confidence is relatively lower and more uniformly distributed. This phenomenon indicates a strong positive correlation between the model's internal confidence and its prediction accuracy, reflecting a decision-making mechanism with good interpretability and calibration. However, the confidence distribution pattern for the Floor hierarchy differs from other hierarchies: both correct and incorrect predictions exhibit right-skewed distributions that are relatively similar. This suggests that the model's deterministic judgment in this hierarchy is less distinct than that of others. A plausible explanation is that, within the underlying behavioral logic of LLM agents, the specific floor selection within a given building category lacks strong discriminative power or is not a core determinant of energy-related behavior. For instance, when fulfilling "experiment" activity requirements, selecting laboratories on different floors within the same building may be considered reasonable. Consequently, the model fails to develop strong discriminative patterns for floor features during learning, resulting in distinct confidence distribution characteristics within this hierarchy.

Furthermore, Figure 5(h) reveals the macro-level relationship between confidence and accuracy on the validation set. As shown, the prediction accuracy of the samples increases monotonically as the model's confidence rises. This important finding provides a crucial foundation for constructing an efficient and energy-saving hybrid simulation system. In practical large-scale simulation applications, an integrated mechanism driven collaboratively by the deep learning model and LLM can be designed: when the deep learning model's prediction confidence for a particular behavior sequence exceeds a preset threshold (e.g., 0.8), its prediction is directly adopted; whereas when the confidence is low, the computationally expensive but reliably accurate LLM is invoked for decision-making. Such a dynamic routing strategy holds promise for maintaining high overall simulation accuracy while substantially reducing the frequency of LLM invocations. This significantly conserves computational resources and offers a practical technical pathway to address the resource constraints of LLM in large-scale simulations.

Conclusion

This paper successfully designs and validates a deep learning framework that integrates an MLP encoder and a Transformer decoder to efficiently generate last-mile energy behaviors of individuals within communities. By learning from simulation data generated by limited-scale LLM agents, the model accurately fits complex energy-consuming behavior sequences, providing a viable technical pathway to address the high computational costs associated with LLMs in large-scale agent-based modeling simulations. The main conclusions of this study are as follows:

1. An efficient deep learning architecture for behavior generation is proposed. To precisely model the LLM Agent's behavior, the deep learning architecture employs an MLP encoder to encode heterogeneous Agent features into a high-dimensional representation, followed by a Transformer decoder for hierarchical generation of behavior sequences.
2. Bayesian optimization is introduced for automated hyperparameter tuning, achieving an average accuracy of 0.8881 and an overall accuracy of 0.7576 on the validation set. This performance is comparable to that of a simulation driven solely

by an LLM, demonstrating its viability as an efficient alternative.

3. Confidence analysis reveals the model's reliable decision-making mechanism. Furthermore, the model's prediction accuracy increases significantly under high-confidence conditions (>0.8), providing a basis for constructing a "Deep Learning-LLM" hybrid simulation system. This approach can further optimize computational resource allocation while ensuring simulation reliability.

Future Work

Building on the deep learning framework proposed in this study, future work can extend its application to energy consumption management and optimal energy dispatch in energy-transportation coupled systems. The model can efficiently and accurately generate individual last-mile energy behavior sequences, providing a novel technical pathway for system-level dynamic energy dispatch and optimization. Specifically, future research directions include:

1. **Fine-grained Energy Consumption Prediction and Dynamic Management.** Based on the high-resolution, individual-level energy behavior sequences generated by this model, a more accurate equipment-level energy demand profile can be constructed compared to traditional statistical or rule-driven methods. Compared with conventional approaches, this method can significantly enhance the spatiotemporal accuracy of short- and ultra-short-term load forecasting and better characterize behavioral uncertainty. In the context of energy-transportation coupling, more accurate predictions enable the system to more reliably integrate electric vehicle charging behaviors, building energy use patterns, and distributed energy resource outputs. Consequently, this integration reduces the safety redundancy capacity in dispatch optimization, increases the proportion of renewable energy consumption, and strengthens the system's adaptability to demand fluctuations.
2. **Responsive Energy Strategies and Personalized Energy-Saving Guidance.** The sequential behavioral data generated by the model can be used to design personalized demand response mechanisms. By identifying the energy usage patterns and flexibility of different user groups, differentiated electricity price incentives or energy-saving prompt strategies can be formulated. This approach improves the precision of user-side responses and overall energy efficiency, demonstrating high application potential particularly in structured scenarios with distinct energy usage patterns, such as campuses and industrial communities.

References

1. Chen, S., Zhang, H., Guan, J. et al., "Agent-based Modeling and Simulation of Stochastic Heat Pump Usage Behavior in Residential Communities," *Building Simulation, Tsinghua University Press* 13 (2020): 803-821.
2. Chingcuanco, F. and Miller, E.J., "A Microsimulation Model of Urban Energy Use: Modelling Residential Space Heating Demand in ILUTE," *Computers, Environment and Urban Systems* 36, no. 2 (2012): 186-194.
3. Tian, S., Lu, Y., Ge, X. et al., "An Agent-Based Modeling Approach Combined with Deep Learning Method in Simulating Household Energy Consumption," *Journal of Building Engineering* 43 (2021): 103210.
4. Ding, Z., Hu, T., Li, M. et al., "Agent-Based Model for Simulating Building Energy Management in Student Residences," *Energy and Buildings* 198 (2019): 11-27.
5. Qiao, Q. and Yunusa-Kaltungo, A., "A Hybrid Agent-Based Machine Learning Method for Human-Centred Energy Consumption Prediction," *Energy and Buildings* 283 (2023): 112797.
6. Zhang, Z., Jing, R., Lin, J. et al., "Combining Agent-Based Residential Demand Modeling with Design Optimization for Integrated Energy Systems Planning and Operation," *Applied Energy* 263 (2020): 114623.
7. Chang, Y., Wang, X., Wang, J. et al., "A Survey on Evaluation of Large Language Models," *ACM Transactions on Intelligent Systems and Technology* 15, no. 3 (2024): 1-45.
8. Thirunavukarasu, A.J., Ting, D.S.J., Elangovan, K. et al., "Large Language Models in Medicine," *Nature Medicine* 29, no. 8 (2023): 1930-1940.
9. Cheng, M., Piccardi, T., and Yang, D., "CoMPosT: Characterizing and Evaluating Caricature in LLM Simulations," arXiv preprint arXiv:2310.11501, 2023.
10. Veselovsky, V., Ribeiro, M.H., Arora, A. et al., "Generating Faithful Synthetic Data with Large Language Models: A Case Study in Computational Social Science," arXiv preprint arXiv:2305.15041, 2023.
11. Brand, J., Israeli, A., and Ngwe, D., "Using GPT for Market Research," *Harvard Business School Marketing Unit Working Paper No. 23-062* (2023).
12. Argyle, L.P., Busby, E.C., Fulda, N. et al., "Out of One, Many: Using Language Models to Simulate Human Samples," *Political Analysis* 31, no. 3 (2023): 337-351.
13. Gao, C., Lan, X., Lu, Z. et al., "S3: Social-Network Simulation System with Large Language Model-Empowered Agents," arXiv preprint arXiv:2307.14984, 2023.
14. Horton, J.J., "Large Language Models as Simulated Economic Agents: What can we Learn from Homo Silicus?" *National Bureau of Economic Research* (2023).
15. Aher, G.V., Arriaga, R.I., and Kalai, A.T., "Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies," in *Proceedings of the*

International Conference on Machine Learning, PMLR, 2023, 337-371.

16. Brookins, P. and DeBacker, J.M., "Playing Games with GPT: what Can we Learn About a Large Language Model from Canonical Strategic Games?" *SSRN preprint 4493398* (2023).
17. Pinkus, A., "Approximation Theory of the MLP Model in Neural Networks," *Acta Numerica* 8 (1999): 143-195.
18. Tolstikhin, I.O., Houlsby, N., Kolesnikov, A. et al., "MLP-Mixer: An all-MLP Architecture for Vision," *Advances in Neural Information Processing Systems* 34 (2021): 24261-24272.
19. Vaswani, A., Shazeer, N., Parmar, N. et al., "Attention is All You Need," *Advances in Neural Information Processing Systems* 30 (2017).
20. Han, K., Xiao, A., Wu, E. et al., "Transformer in Transformer," *Advances in Neural Information Processing Systems* 34 (2021): 15908-15919.
21. Han, K., Wang, Y., Chen, H. et al., "A Survey on Vision Transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, no. 1 (2022): 87-110.
22. Jing, H., Hu, J., Ou, S.S. et al., "A Data-Driven and Physics-Based Model for Assessing Real-World Usage Behavior Impacts on Electric Vehicle Battery Life," *Journal of Energy Storage* 144 (2026): 119680.
23. Jing, H., Hu, J., Ou, S.S. et al., "Scalable and Generalizable Deep Learning for Battery State of Health Estimation in On-Road Electric Vehicles," *Journal of Energy Chemistry* (2025).
24. Eggensperger, K., Feurer, M., Hutter, F. et al., "Towards an Empirical Foundation for Assessing Bayesian Optimization of Hyperparameters," *NIPS Workshop on Bayesian Optimization in Theory and Practice* 10, no. 3 (2013): 1-5.
25. Lindauer, M., Eggensperger, K., Feurer, M. et al., "SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization," *Journal of Machine Learning Research* 23, no. 54 (2022): 1-9.
26. Albahli, S., "Efficient Hyperparameter Tuning for Predicting Student Performance with Bayesian Optimization," *Multimedia Tools and Applications* 83, no. 17 (2024): 52711-52735.

Contact Information

Shiqi(Shawn) Ou
sou@scut.edu.cn; oushiqi@pazhoulab.cn
Phone number: +86-020-81181684
Mailing address:
1. South China University of Technology, School of Future Technology, 777 Xingye Ave East, Panyu District, Guangzhou, Guangdong, 511442, China
2. Guangdong Artificial Intelligence and Digital Economy Laboratory (Guangzhou), 70 Yuean Road, Haizhou District, Guangzhou, Guangdong 510335, China.

Acknowledgments

This research is supported by the Introduced Innovative R&D Team of Guangdong (2023ZT10L145). All opinions expressed in this paper are the authors' and do not necessarily reflect the policies and views of sponsors.

ISSN 0148-7191