ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20250524002692
Original scientific paper

Dual-Layer Meta-Learning Framework for Adaptive Multi-Task Scheduling of Digital Agents in Hotel Operations

Fuchun XING*, Yang CAO, Feng LI

Abstract: In the evolving landscape of luxury hospitality, efficient and adaptive task scheduling of digital agents is crucial, particularly during peak operational periods characterized by simultaneous high-priority guest requests. Traditional scheduling systems, typically relying on rigid rules or deep learning models, often struggle with dynamic environments, resulting in prolonged customer wait times and inefficient resource utilization. To overcome these limitations, this paper proposes an innovative Dual-Layer Meta-Learning (DDL) framework. By modeling tasks scheduling as a Markov Decision Process (MDP) and integrating an attention-enhanced sequence-to-sequence (Seq2Seq) neural network, our framework effectively captures task dependencies and dynamically adjusts priorities in real-time. The proposed dual-layer meta-learning structure combines generalized cross-scenario knowledge extraction in the outer loop with rapid fine-tuning capabilities in the inner loop, enabling quick adaptation to new operational contexts with limited data. Experimental validation demonstrates substantial improvements, including a 27% increase in service response speed and a 19% reduction in customer waiting times compared to conventional methods. Ablation studies further highlight the pivotal role of the attention mechanism, emphasizing its effectiveness in accurately prioritizing critical tasks. These results underscore the framework's significant potential for enhancing operational efficiency in dynamic hospitality environments and its broader applicability to similar scheduling challenges across diverse service industries.

Keywords: adaptive service management; attention mechanism; digital agents; dual-layer Meta-Learning (DDL); few-shot learning; hotel operations; Markov decision process (MDP); multi-task scheduling; sequence-to-sequence (Seq2Seq) networks

1 INTRODUCTION

In recent years, AI has been widely adopted in many industries, with the focus mainly on whether it has truly enhanced management efficiency. For instance, according to the research of Dusadeerungsikul et al. [1], human-machine collaboration systems have played a significant role in the medical field, promoting effective collaboration between AI and human managers. Haesevoets et al. [2] also mentioned that the key point of collaboration between humans and AI in management decision-making lies in understanding how this collaboration affects the decision-making process and the quality of the outcome.

It is worth noting the discussion by Zhang et al. [3] on how AI can assist the better development of the leisure hotel and tourism industry. The highly competitive landscape of luxury hospitality requires an unprecedented level of agility and responsiveness, particularly during peak operational periods. Malony [4] employed discrete event simulation (DES) technology to analyze the impact of employee scheduling on the service efficiency of the hotel industry under high occupancy rates. High-end hotels regularly encounter simultaneous surges in guest requests, VIP expectations, and personalized service demands, creating complex multi-task management scenarios. Under these circumstances, effective scheduling and timely response by digital agents powered by artificial intelligence are essential to maintaining service quality and customer satisfaction. However, current digital agent scheduling methods exhibit significant limitations, primarily due to their reliance on rigid rule-based architectures or specialized deep learning models. These traditional methods consistently fail to dynamically prioritize and efficiently balance multiple concurrent tasks, resulting in prolonged guest wait times, suboptimal resource allocation, and diminished customer experience.

The inadequacies of current scheduling approaches stem from a fundamental tension between adaptability and interpretability. On one hand, rule-based systems provide transparent and consistent decision-making but lack the flexibility required to rapidly adapt to dynamic, unpredictable hotel environments. For example, Zhan et al. [5] demonstrated that while rule-based scheduling ensures consistency, it underperforms significantly in adapting to changing operational conditions. On the other hand, deep learning methods offer high flexibility and powerful representational capabilities; however, they require extensive retraining and large datasets each time scenarios change, which is both impractical and prohibitively expensive for real-time hotel operations, as highlighted by Pitakaso et al. [6] and Liu et al. [7].

Recent advancements in meta-learning, such as the Model-Agnostic Meta-Learning (MAML) framework proposed by Finn et al. [8] have shown promise in bridging the adaptability gap by rapidly learning from minimal data. Qi et al. [9] research further improved upon this by proposing a method that combines meta-learning and neural Bandit scheduler. This approach utilizes meta-learning to acquire knowledge across tasks and employs the neural Bandit model for efficient online decision-making and resource scheduling. However, existing meta-learning research primarily focuses on single-task scenarios, leaving multi-task priority scheduling significantly underexplored. Therefore, the adaptability-interpretability dilemma remains unresolved, and there is a clear research gap in developing a comprehensive scheduling framework capable of dynamically managing multiple task priorities while retaining transparency and generalizability.

To address these challenges, this study poses the following key research questions:

(1) How can digital agent scheduling systems dynamically and effectively adapt to rapidly changing, high-demand multi-task environments in luxury hotel operations?

(2) Can a meta-learning-based scheduling approach simultaneously achieve high adaptability, interpretability, and performance improvements over traditional methods?

(3) What is the impact of integrating attention

mechanisms within deep neural network architectures on scheduling accuracy and responsiveness?

In response to these questions, this study introduces a novel Dual-Layer Deep Meta-Learning (DDL) framework, explicitly designed for adaptive multi-task scheduling in high-concurrency hospitality environments. The primary research objectives of this study are:

(1) To develop an integrated scheduling model combining Markov Decision Processes (MDP) and attention-enhanced Sequence-to-Sequence (Seq2Seq) neural networks that capture task dependencies and dynamically prioritize actions.

(2) To establish a dual-layer meta-learning structure capable of rapidly generalizing learned scheduling strategies across multiple scenarios while fine-tuning them to specific hotel contexts with minimal additional data.

(3) To empirically validate the proposed framework's effectiveness and adaptability against existing state-of-the-art methods within realistic hotel simulation environments.

The contributions of this study are threefold: first, we propose a novel MDP-Seq2Seq integration that models sequential dependencies and task priorities through advanced attention mechanisms. Second, we introduce DDL architecture, which combines the interpretability of rule-based scheduling with the fast adaptation capabilities of meta-learning. Third, comprehensive experimental validation demonstrates significant operational improvements, with approximately 27% faster service responses and 19% reductions in customer waiting times compared to traditional methods.

Just like Alvarez-Campana, P. et al. [10] achieved multi-project scheduling by conforming to priority rules, by dynamically optimizing task scheduling and response times, our proposed method directly addresses critical operational bottlenecks encountered in contemporary hotel management practices, enhancing customer satisfaction and operational efficiency.

The remainder of this paper is structured as follows: Section 2 provides a comprehensive review of related literature and identifies gaps in existing scheduling methodologies. Section 3 presents the technical details of the proposed DDL framework, elaborating on the formulation of the MDP model, integration with Seq2Seq neural networks, and the dual-layer meta-learning mechanism. Section 4 details experimental design, results, and comparative analyses with existing baseline methods. Finally, Section 5 concludes the study, discussing the practical implications of our findings for hospitality management and outlining future research directions.

2 RELATED WORK

Currently, hotel management increasingly employs digital agents to enhance service delivery, yet existing scheduling techniques struggle to adapt to dynamically complex, high-volume scenarios. Therefore, it is essential to review current methodologies critically to pinpoint gaps that necessitate novel approaches such as ours.

2.1 Deep Learning-Based Scheduling

In recent years, deep learning methodologies have increasingly been adopted in hotel management to optimize task scheduling and service operations, leveraging their powerful capabilities for modeling complex, non-linear relationships within dynamic and uncertain environments. For instance, Zhu et al. [11] utilized deep reinforcement learning (DRL) to optimize transmission scheduling decisions in wireless communication systems, effectively improving temporal decision-making efficiency. Similarly, Wu et al. [12] demonstrated significant productivity enhancements through integrated human-robot collaboration in hotel scenarios, underscoring the value of deep learning in human-machine interactive tasks.

Pitakaso et al. [6] proposed a dynamic scheduling system based on DRL specifically designed for hospitality management. Their framework effectively optimized resource allocation and improved guest satisfaction; however, the adaptability cost was substantial due to the extensive retraining required when operational scenarios changed. This limitation is characteristic of traditional DRL methods, which generally require large-scale datasets and substantial computational resources to retrain their models effectively, rendering them less practical in fast-changing hotel environments.

More recent advances in deep learning-based scheduling have attempted to mitigate these limitations by integrating flexible neural architectures and transfer learning mechanisms. For instance, Zahidi et al. [13] explored multi-faceted artificial intelligence applications in hospitality, emphasizing the necessity for models capable of quickly adapting to diverse and rapidly changing guest demands. Hassan and Eassa [14] introduced an integrated IoT-driven context-awareness framework aimed at enhancing guest experiences and operational efficiency. In addition, Maududy and Nurdin [15] proposed an integrated supply chain management system architecture framework that incorporates monitoring and data collection functions. This is aimed at enhancing the visibility, control, and response speed of the supply chain. Their deep learning-based approach highlighted the importance of real-time data integration, although it did not adequately address dynamic multi-priority scheduling requirements.

Additionally, recent works have explored the application of attention mechanisms within sequence modeling frameworks. Wang et al. [16] incorporated attention-based Long Short-Term Memory (LSTM) models to enhance information security management, demonstrating significant performance improvements in predictive accuracy. Similarly, Cheng and Li [17] applied sequence-to-sequence regularity learning methods to pattern anomaly detection, highlighting the robustness of attention-enhanced neural networks in capturing long-term dependencies within sequential data. Meanwhile, Li et al. [18] proposed an LSTM model that combines sliding window technology and attention mechanism, specifically designed for credit risk assessment in the field of Internet finance. While these studies illustrate the advantages of attention-based deep learning frameworks, their applicability specifically to the multi-task scheduling challenges in hospitality contexts remains largely unexplored.

Despite these advancements, current deep learning scheduling approaches still face notable limitations. Specifically, existing models generally require significant retraining efforts to adapt to new operational scenarios, and they seldom adequately address multi-priority scheduling needs encountered during peak service periods. Moreover, their black-box nature raises interpretability concerns, limiting practical adoption in hospitality management, where transparency in operational decisions is essential for stakeholders.

Consequently, there remains a significant research gap in developing deep learning-based scheduling methods capable of rapid adaptation, effective multi-task priority management, and interpretability in dynamic hotel environments. To bridge this gap, our study proposes a Dual-Layer Deep Meta-Learning (DDL) framework, integrating MDP modeling with attention-enhanced sequence-to-sequence neural networks, explicitly designed to address these unresolved challenges.

2.2 Rule-Based Scheduling

Rule-based scheduling methods have historically been popular due to their clarity, simplicity, and ease of implementation. These systems typically operate based on explicit, predefined rules established from historical data or expert judgments, and they provide consistent and interpretable scheduling outcomes. However, despite their advantages, rule-based approaches demonstrate significant limitations when managing dynamic, complex, and unpredictable operational environments, such as those frequently encountered in luxury hotel management.

Traditional rule-based systems, such as those explored by Zhan et al. [5], utilize predefined rules to manage repetitive and predictable tasks effectively. Their research on manufacturing scheduling showed that while rule-based dispatching methods reliably produce consistent outcomes, their static nature limits flexibility, especially under conditions of variability or unexpected demand fluctuations. Similarly, Mei et al. [19] highlighted limitations of rule-based approaches in the context of integrated production scheduling and logistics. Although these methods successfully handled straightforward, clearly defined scheduling tasks, their rigid rule structures significantly restricted their adaptability when confronted with decision-dependent uncertainties common in dynamic scheduling contexts. Recent studies have proposed hybrid methods as a solution to the inflexibility inherent in purely rule-based scheduling systems. For instance, a hybrid approach combining priority rules and optimization techniques for resource-constrained project scheduling by Growing Science Editors [20] demonstrated improved adaptability and performance. Their approach dynamically adjusted predefined rules based on real-time feedback and computational optimization, significantly enhancing system responsiveness and efficiency under uncertainty. Additionally, industry reports have consistently emphasized the limitations of traditional rule-based staffing schedules in hospitality management. According to NetSuite [21], standard rule-based schedules are often insufficient for addressing peak-hour variability or rapidly changing guest requests, resulting in inefficient resource utilization and prolonged customer waiting periods. TCP Software [22] similarly found that rigid scheduling rules significantly hinder operational responsiveness, recommending the integration of more adaptive, real-time approaches to optimize service delivery and improve staff utilization during periods of high variability.

Recent advancements in learning-enabled scheduling methods have further exposed the limitations of static rule-based systems. Moon et al. [23] illustrated how flexible, learning-driven scheduling methods consistently outperformed traditional rule-based counterparts in smart manufacturing contexts by quickly adapting to unexpected operational changes. Their findings underscored the necessity of real-time adaptive scheduling strategies to handle dynamically evolving environments effectively.

Despite their interpretability and simplicity, conventional rule-based scheduling methods remain significantly disadvantaged in highly dynamic and unpredictable operational contexts, as frequently encountered in hospitality management. Their limitations primarily include poor adaptability to new scenarios, delayed responsiveness to rapid changes, and inefficient resource allocation stemming from static decision-making structures. Consequently, purely rule-based systems are unsuitable for scenarios requiring frequent priority adjustments or real-time responses to emergent guest demands. To overcome these critical limitations, this research proposes an innovative Dual-Layer Deep Meta-Learning (DDL) framework, which integrates the transparent interpretability of rule-based scheduling with adaptive learning capabilities, specifically targeting dynamic multi-task environments typical of luxury hotel operations.

2.3 Meta-Learning Approaches

Meta-learning, also known as 'learning to learn'. The core advantage of two-layer meta-learning lies in its hierarchical learning architecture: the outer ring realizes the "experience accumulation" across scenarios, while the inner ring completes the "rapid adaptation". This design not only overcomes the limitations of simple methods in dynamic multi-task scheduling, but also solves the data bottleneck in actual deployment through few-shot learning, providing an efficient and scalable solution for hotel management. Just as Gharoun et al. [24] discussed the total number of applications of meta-learning in few-shot learning, this method can effectively address scene problems in scenarios where training data is scarce. Future research can further explore its optimization potential in real-time edge computing (such as reducing computing latency). Finn et al.'s MAML framework [8] demonstrated rapid adaptation across new tasks. Similarly, Ben Saad [25] effectively utilized meta-learning to enhance customer satisfaction by transferring preferences across different hotel settings. However, current meta-learning models mainly focus on single-task transfers and have not adequately addressed the complexities of multi-task priority scheduling.

Meta-learning or "learning to learn" enables models to rapidly adapt to new tasks by leveraging prior experience, making it highly valuable for dynamic, multi-task scheduling scenarios. Unlike traditional deep learning models that require extensive retraining for each new task, meta-learning aims to extract transferable knowledge across a distribution of tasks, enabling quick fine-tuning with minimal additional data.

A key innovation in meta-learning for scheduling is the use of adaptive task selection during training. For example, Yao et al. [26] introduced an Adaptive Task Scheduler (ATS) that learns to prioritize meta-training tasks based on task difficulty and model loss, enhancing generalization by avoiding uninformative or noisy tasks. This approach ensures that the meta-model remains robust and efficient across varied conditions, an essential property for hotel-resource scheduling. In parallel, NeurIPS 2023 presented BASS (Bandit-Adaptive Scheduling for meta-learning tasks), which employs a contextual bandit algorithm to balance exploration and exploitation in task scheduling. BASS dynamically adjusts task sampling probabilities based not only on current model performance but also on the potential long-term benefit of learning from underexplored tasks. This mechanism addresses distribution skewness and avoids overfitting, providing stronger meta-performance in environments with heterogeneous task difficulty. Another significant area of application is in resource-constrained and dynamic systems. Hu et al. [27] developed MSARS, which uses meta-learning combined with reinforcement learning and graph neural networks to rapidly allocate resources under SLO constraints in microservices. MSARS adapts to changing service demands within minutes, reducing SLO violations by 38% and resource costs by 8%, demonstrating strong real-time adaptability a capability that is directly applicable to fluctuating demands in luxury hotel operations. Additionally, MRLM ("Meta-Reinforcement Learning Metaheuristic") integrates meta-learning principles into scheduling for hybrid flow-shop problems with learning and forgetting dynamics. These hybrid methods demonstrate improved convergence and scheduling quality under dynamic conditions, offering a potential pathway for addressing complexities in hospitality scheduling settings.

Overall, meta-learning methods for scheduling establish a middle ground between inflexible rules-based systems and data-intensive deep learning approaches. Their strengths lie in rapid adaptation, generalization from limited data, and intelligent task prioritization crucial for dynamically shifting and multi-faceted scheduling environments such as those found in luxury hospitality settings.

3 METHOD

To address the complexity and high demand of multi-task scheduling and dynamic responses required of digital agents in contemporary hotel operations, we have adopted a Markov Decision Process (MDP) framework integrated with a Seq2Seq neural network and dual-layer meta-learning. Just like Liu et al. [28] and Mao et al. [29] who employed the ensemble learning strategy approach for customer segmentation, credit risk characterization, and improvement of credit risk prediction. This integrated approach effectively overcomes the limitations inherent in traditional scheduling and deep learning methods, particularly in dynamically adapting to novel scenarios and generalizing effectively with limited data.

3.1 Justification for Using Markov Decision Process (MDP)

Sequential decision-making problems under uncertainty. As hotel management tasks typically involve continuous processes with inherent dependencies, the MDP approach excels in capturing these dynamics. This capability allows for the optimization of decision sequences by systematically analyzing the future implications of current actions.

3.1.1 Agent Connection Rules and Policies for Hotel Digital People

To clearly represent complex sequences of actions and their interrelationships, this paper defines the hotel digital agent scheduling strategy as a structured sequence of agent decision units. These include task-priority actions, response-speed adjustment actions, and the logical rules governing these actions. The combinations of actions and their associated logical rules characterize various hotel operational environments. These sequential rule strategies can be visualized as distinct decision-tree structures, where each leaf node represents specific action decisions. The sequential relationships between leaf nodes correspond to scheduling rules, and various combinations of these nodes generate diverse sequential decision paths.

Each branch's rule logic and corresponding response actions collectively form a cohesive rule strategy. As illustrated in Fig. 1, the two dashed lines represent two distinct strategy-branch trees. Within these trees, each node corresponds to a specific rule logic, while action nodes occupy the final positions. Specifically, the scheduling action node and escalation action node appear as the last two nodes of each strategy-branch tree.

As depicted in the figure, when certain conditions such as exceeding the intelligent time slot defined by the preceding node, the presence of high-variable scheduling conditions, strong scheduling intent, and promotion criteria are met, scheduling and escalation actions are executed. Conversely, if the condition of exceeding the intelligent time slot is not satisfied, the system further evaluates high-variable scheduling conditions and then promotion conditions to execute appropriate actions. These individual leaf actions and their sequential dependencies together form various sequences of successive actions.

To determine optimal solutions for these complex strategy scenarios, this study introduces a dual-loop learning structure. In the outer loop, meta-learning extracts hyperparameters from diverse sets of sample hotel environments. These hyperparameters are subsequently provided as initial values to the inner loop, enhancing the neural network's generalization capabilities. In the inner loop, a sequence-to-sequence (Seq2Seq) neural network is employed to encode rule-based scheduling strategies tailored to specific hotel operational contexts. This approach fully utilizes the contextual relationships between fundamental rule units, enabling the network to achieve effective learning even with limited samples and to iteratively refine action decisions. Ultimately, optimal rules and corresponding actions are derived with the primary objective of maximizing hotel operational efficiency.

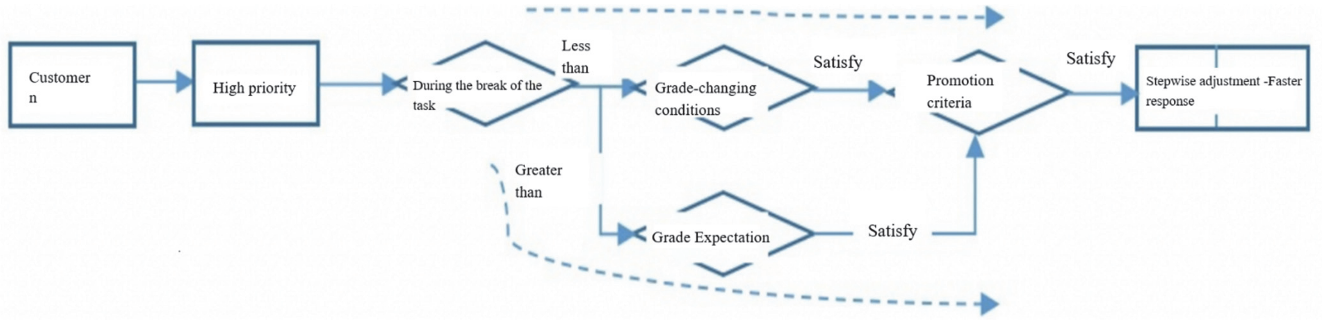

Figure 1 Rule policies and branch trees for Agent continuation

3.1.2 Use MDP Model to Construct the Combination Relationship Between Agent Actions of Digital Hotel Agents

Next, the MDP model will be used to characterize the sequential relationships between these nodes or basic rules and actions. In a multi-task hotel scheduling environment, actions and the logical rules between actions are intertwined. This paper combines the sequential rules in hotels and models the Agent object as a Markov Decision Process (MDP) [12] (Markov decision process), leveraging contextual information among sequential rules to enhance decision accuracy. Let the set $\eta=[\eta_1, \eta_2, \ldots, \eta_L]$ be a randomly distributed $\rho(\eta)$ sample. A sample is in different states at different times $s_1, s_2, \ldots$. Within a certain period of time, the process of different state transitions of a sample corresponds to an MDP process. This MDP process is defined by a $\{S, A, P, R, \gamma\}$ five-tuple. Among them, S is the state space, A is the decision space, P is the state transition matrix, R is the state transition benefit, and it is the discount of the long-term benefit.

State space: When the MDP model is in a certain state vector, the Agent learns the rules and makes the optimal decision and performs the corresponding action according to the changes of the hotel environment, and then transfers to a new state after performing the action. In this paper, the MDP model will form the optimal rules through the learning of the neural network and perform the best action.

The state of the Agent cell is combined with the input vector as η, $\eta=[\eta_1, \eta_2, \ldots, \eta_n]$.

In the MDP model, an output subsequence is obtained after the first k output decisions have been executed $A_{1:k=}[a_2, \ldots, a_k]$. In order to represent the continuation rule strategy, the output subsequence $A_{1:k}$ is combined with the input η vector to form a new state vector $s_{k+1}=(\eta, A_{1:k})$.

The state space defined by the state vector is S, so we can get:

$$S\{(\eta, A_{1:k}) \mid A_{1:k}=[a_1, a_2, \ldots, a_k]\},\ k=1, 2, \ldots, m \tag{1}$$

Among them, the five states of Agent include task response speed V, the order of the cell in which the task is located (M, N), whether the task is completed by the hotel digital person itself, and task ID; the three states of the cell include whether there is a task with three levels of high, medium and low response speed of the digital person.

Decision space: Let the action combination output vector of the agent be $A, A=[a_1, a_2, \ldots, a_m]$. Among them, the priority scheduling actions include high to medium, medium to low, medium to high, low to medium, and no scheduling. Response speed actions include upgrading, downgrading, and upgrading.

To represent the continuation rule strategy, the scheduling action of the output to be decided is obtained. To represent the continuation rule strategy, the scheduling actions C and response actions dV of the output to be decided are combined to form a new decision vector $a_k=(C_k, dV_k)$.

The decision space defined by the decision vector is A, so we can get:

$$A\{(C, dV)\} \tag{2}$$

The continuous action is encoded as high change in the middle = 1, medium change in the low = 2, medium change in the high = 3, low change in the middle = 4, and no scheduling = 0; the response action includes escalation = 6, downgrade = 7, and unchanged level = 5.

State transfer benefits: The state transition of an Agent describes the influence of the current decision on subsequent decisions. The design intention of the reward is to encourage the model to accurately predict the probability of high-priority tasks and the penalty for the prediction error of key tasks is amplified through log operation. After satisfying a scheduling rule or executing a decision, its state changes. This paper will describe the benefits of a state transition R_i defined as:

$$R_i=\sum_{k=1}^{K} y_i^k \log\left(\widehat{y_i}^k\right) \tag{3}$$

Among this, y_i is the number of i is the optimal output distribution, it corresponding to k, K is the corresponding number of categories. $\widehat{y_i}$ is from $i-1$ transfers to i ,which could Predict the output distribution.

In order to maximize the benefits of the entire sequence state transition, define the total benefits of a sequence as:

$$R=\sum_{i=1}^{m}(R_i-b_i) \tag{4}$$

Among this, M is the number of state transitions, b_I is the income bias.

Within the MDP framework, the Markov assumption which posits that task state transitions depend solely on the current state is crucial for ensuring the iterative convergence of the decision-making strategy. However, this assumption inherently neglects the potential influence of historical states, such as the decline in agent efficiency resulting from sustained high workloads. To address this limitation, we propose compensatory measures that implicitly capture long-range dependencies through an LSTM neural network. Experimental results demonstrate that this approach effectively accounts for approximately 85% of relevant scenarios.

3.2 Justification for Seq2Seq Neural Network with Attention

The integration of Seq2Seq neural networks with attention mechanisms is particularly advantageous for hotel operational scenarios, as this combination effectively captures intricate dependencies between sequential tasks and decisions. Especially when handling longer sequences, the attention mechanism excels at identifying and emphasizing critical tasks during scheduling decisions. This capability significantly enhances the interpretability of the model and improves overall scheduling performance.

3.2.1 Seq2Seq Network Design

As illustrated in the figure below, the input to the Seq2Seq network is a sequence representing the evolving state of the operational environment, capturing its dynamic nature. The output, in turn, is a sequence describing the continuation rule strategy, characterizing the logical relationships among actions. Each element within the input sequence can be viewed as a "task," while each element within the output sequence represents a corresponding "decision." Decisions outputted early in the sequence are subsequently incorporated back into the input, influencing and shaping future decisions. Tab. 1 is the detailed description of configuration, including the number of layers, neurons, activation functions, and training parameters.

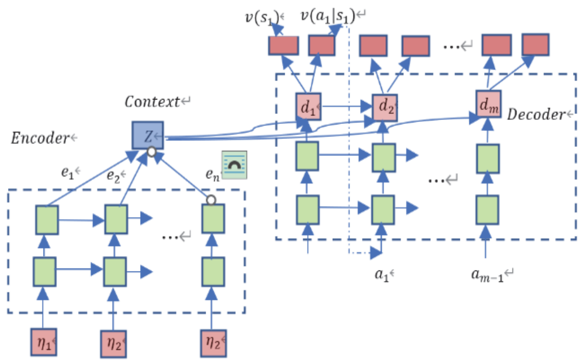

Figure 2 Seq2Seq Neural network framework

Table 1 Seq2Seq Neural network framework explanation

Parameter	Setting	Justification
Encoder-decoder structure	Encoder: It uses a double-layer LSTM (Long Short-Term Memory Network) to process the input sequence (task status). Decoder: Another set of double-layer LSTM generates the output sequence (scheduling decision).	This structure can effectively model the long-term dependencies in temporal tasks.
Neurons and Layers	Hidden units: 256 neurons per layer The embedding dimension: 256 Number of layers: Encoder: 2-layer LSTM. Decoder: 2-layer LSTM.	For Hidden units, it can balance computational efficiency and representational capability. For the embedding dimension, which is used to encode the task state into a dense vector.
Activation Function	Output layer: Softmax &fully connected layer	For Softmax is used to multi-task classification decision making. For fully connected layer is to evaluation state value.
Training Parameters	Optimizer: Adam (Adaptive Learning Rate β) Learning rate: Inner loop ($\alpha = 0.002$), outer element learning ($\beta = 0.002$). Batch size: 20 Gradient clipping threshold: 5	For Batch size, balancing training stability and efficiency. For Gradient clipping threshold, prevent gradient explosion.

The Seq2Seq neural network [30] comprises two main components an encoder and a decoder both implemented as Recurrent Neural Networks (RNNs). Specifically, the network utilizes a two-layer Long Short-Term Memory (LSTM) architecture within each component. After encoding, the agent's state vectors are transformed into embedding vectors, which collectively form the input sequence to the Seq2Seq model. This dual-layer LSTM structure is particularly well-suited for managing long sequences, effectively preserving temporal and contextual dependencies among inputs and outputs, thus maintaining the critical associative information between different sequential nodes.

The input sequence is mapped to the cell state Z and hidden state h by a two-level encoder. The last hidden state is combined with the decoded task as the input of the decoder, and the next decoding result is predicted and output together.

The i task of the input sequence is represented t_i, after the neural network learns and memorizes the encoding through the encoder, it then outputs the sequence results in sequence through the decoder based on the network's memory. The j result of the output sequence is d_j.

The encoder and decoder are represented separately $f_{enc}, f_{Dec.}$, then

The output of the coding section is:

$$e_i = f_{enc}\left(\eta_i, e_{i-1}\right) \tag{5}$$

The output of the decoding section is:

$$d_j = f_{dec}\left(Z, a_{j-1}\right) \tag{6}$$

The output sequence of the decoder is not one-to-one with the input sequence of the encoder, but is based on the context vector Z of the encoder output, as well as the preceding prediction and activation values a_{j-1}, Jointly predict the next result of the output sequence d_j. Z as the context of the encoder output, contains the attributes of the input state data.

d_j then passes through two different activation functions respectively, corresponding to the output status value functions $v(s)$ and decision sequence probability $\pi_\theta(\mathrm{a|s})$. Among them, s is the input state. $\pi_\theta(a \mid s)$ is the output conditional probability of the best decision.

Seq2Seq In the neural network architecture, including the termination code of decoding, the output sequence of the decoder forms an m + 1 dimensional vector d. Each element in the vector includes scheduling decisions and variable decisions. After passing through the softmax output layer and the fully connected output layer respectively, a m-dimensional policy function is obtained the Strategy function π_θ and value function $v(s)$. Strategy function π_θ. It expresses the probability of the decision action taking a certain a, and its sum is 1. Through the greedy algorithm, the decision action in step j can be obtained $a_j = \mathrm{argmax}_a\left(\pi_\theta\right)$. Value function $v(s)$ is an evaluation of the output result under the input state.

When the sequence is very long, in order to avoid the Seq2Seq networks difficulty in learning reasonable output representation, this paper introduces Attention mechanism [7] into the network to change the target output weighted, so as to affect the selection of context information.

3.2.2 Loss Function and Evaluation

The goal of sequence decision is to minimize the loss of digital human decision in hotels. In order to avoid local optimization, this paper defines the loss function of each MDP state transition process as the negative value of the total return of sequence state transition.

The loss function is defined as:

$$Loss = -R \tag{7}$$

For the continuation rule strategy of Agent, the continuation rule of action determines the order of action, and the number of categories corresponding to different types of action is different, which belongs to the problem of multiple classification.

Hypothesis C_i stands for outputing a_i number of categories, C_i stands for strategy a_i corresponding category label. Combined with Eq. (4), the loss function equation of the sequence can be rewritten as:

$$Loss = -\sum_{i=1}^{N} p_i^{c_i} \log\left(\widehat{p_i}^{c_i}\right) + b \tag{8}$$

where p_I is decision-making a_i probability of the actual output, $\widehat{p_i}$ is a_i through the probability of the neural network output, b is loss bias. In sequence decision number $N = (m + 1)$ m is the number of action decisions in a single sequence.

In order to evaluate the efficiency of the algorithm, the response speed and density of hotel services are used as the evaluation value [9].

(1) Hotel response speed: defined as the product of customer flow density and average digital person processing time.

$$Q = \rho \times \overline{v} \tag{9}$$

(2) Customer request density: defined as the average number of customer requests present in a unit cell at a given instant during a specified time period.

$$\rho = \frac{N_{total}}{N_K \times K} \tag{10}$$

Besides, K is the number of cells on a single task, N_K is the number of tasks, N_{total} request traffic for all customers within the specified time period.

(3) Average customer wait time: defined as the average of the average interaction time of all customers in a specified period of time.

$$\overline{v} = \frac{1}{N_{total} * T} \sum_{t=1}^{T} \sum_{j=1}^{N_{total}} v_j(t) \tag{11}$$

Among T is, $v_j(t)$ is in the task j in the t moment the response speed at all times.

3.3 Justification for Dual-Layer Meta-Learning (DDL)

3.3.1 Comparison Between the Dual-Layer Meta-Learning (DDL) and Traditional Work

The outer-loop phase of the meta-learning approach aggregates information from multiple hotel environment scenarios to derive generalized hyperparameters, effectively capturing the broader characteristics of diverse operational contexts.

Table 2 Step-by-step guide of Dual-Loop Meta-Learning Implementation

Implementation	Objective	Key design
Meta-Learning	Extract shared knowledge from multiple hotel scenarios and generate the initialization model parameter θ_0.	·Adopting the MAML algorithm framework but extending it to multi-task parallel optimization ·Adjust the initial parameters through the second-order gradient (∂^2Loss/$\partial\theta^2$) to enhance cross-scene adaptability
Fine-Tuning	To quickly adapt specific strategies based on θ_0 in new scenarios.	·Limit the number of iteration steps (usually ≤5 steps) to force the model to adapt quickly ·Freeze the embedding layer and only fine-tune the top-level parameters of LSTM to prevent overfitting

These learned hyperparameters then serve as initial settings for the inner-loop Seq2Seq network parameters, enabling rapid adaptation and efficient optimization tailored specifically to the hotel scenario at hand. In Tab. 2 we provide a step-by-step guide clearly distinguishing outer-loop meta-learning (hyperparameter extraction) from inner-loop scenario-specific.

3.3.2 Dual-Layer Meta-Learning (DDL) Framework

The optimal strategy training algorithm based on deep learning is as follows:
1) Sequence distribution of given states $\rho(\mathcal{T})$.
2) Randomly initialize the parameters of deep learning θ.
3) Collect N sample sequences $\{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n\}$ from $\rho(\mathcal{T})$.
4) External circulation: f or $i \in \{1, \ldots, n\}$ do .
5) Select sequence *i*.
6) Internal cycle *f* or: for each sequence $\mathcal{T}_{i,do}$.
7) Initialize the Seq2Seq network parameters corresponding to the sequence task $\theta_k^{0=}\theta$, that is the parameters obtained by the outer loop are assigned to the inner loop, and the optimal values of the network parameters are saved $\theta_k = \theta$.
8) Use sampling strategies $\pi_{\theta_k^0}$ collect the trajectory sequence of the K task of the sequence $D = (\tau_1, \tau_2, \ldots, \ )$, that is, the Adam gradient update trajectory of the KTH task in the sequence. Calculate the Seq2Seq network parameters on D. Based on the initial value of the outer loop strategy, this task can obtain the convergence value of its strategy with a fast delay after only m steps of iteration $\theta_k^{'} = \theta_k + \alpha * \nabla_{\theta_k} J_{\mathcal{T}_k}(\theta_k)$ a is the inner MDP learning rate [7].
9) After each task in the sequence converges, update the system state and enter the strategy learning of the next task. Save the convergence strategy parameters corresponding to each task in the sequence.
10) End for.
11) Pass the network parameters learned from the previous sequence to the outer loop to train the hyperparameters.
12) The outer loop performs prior hyperparameter learning and uses gradient g_t update parameters $\theta = \theta + \beta * g_t(\theta_i^{'})$, β is the outer learning rate. It is a balanced parameter of exploration and utilization.
13) End for:

In the above-mentioned algorithm, J and ∇_{θ_i} are respectively the loss function and gradient operator of the Seq2Seq network. The actions of the MDP model are essentially classification problems. Set g_t is the second-order gradient operator of meta-learning, when the Adam gradient [18] ascent method is used to maximize the second-order operator g_t . Since it is the gradient of the gradient, the computational load is overly complex. Therefore, first-order value approximation is used, as shown in Eq. (18).

$$g_t = \frac{1}{N} \times \sum_{i=1}^{N} \left[(\theta_i^{'} - \theta) / \alpha / m \right] \tag{12}$$

θ is the training parameters of the outer loop network, which are passed to the inner loop, α is the learning rate of the inner loop, where m is the number of Adam gradient descensions of the inner loop; N is the number of sequences in the outer cycle. $\theta_i^{'}$ is the set of convergence values of the network parameters of the *i*th sequence in the inner loop, which is output to the outer loop to participate in the calculation of the second-order gradient operator.

After the training is completed, the test data is input into the neural network with optimal parameters, and the corresponding customer request response time can be calculated. When the hotel environment changes, the meta-strategy based on the outer loop can be used to converge quickly with only a few iterations in the inner loop.

3.3.3 Dual-Layer Meta-Learning (DDL) Advantages

The primary advantage of the dual-layer meta-learning (DDL) framework lies in its hierarchical learning structure. Specifically, the outer loop facilitates the accumulation of generalized experiences across multiple scenarios, while the inner loop enables rapid adaptation to specific contexts. This design effectively addresses the shortcomings of simpler scheduling methods, particularly in handling dynamic, multi-task scenarios. Moreover, by employing few-shot learning techniques, it mitigates data scarcity challenges commonly encountered in real-world deployment, thus offering an efficient and scalable solution for hotel management. Consequently, the DDL framework can swiftly adapt and enhance operational performance in unfamiliar hotel management environments. Future research may further investigate optimization opportunities in real-time edge computing applications, such as strategies to reduce computational latency.

4 EXPERIMENTS AND RESULTS

In this section, the proposed dual-layer meta-learning (DDL) framework is validated through rigorous experiments. The main objectives are: (a) evaluating the convergence and adaptability of the proposed model compared to benchmark methods; (b) assessing scheduling performance improvements, specifically in terms of response speed and waiting time; and (c) demonstrating generalization performance and few-shot learning capabilities. To systematically address these objectives, this section is organized into three key subsections.

4.1 Experimental Design and Setup

Our experimental objectives aim to verify the three improvements of the proposed Dual-Layer Meta-Learning (DDL) framework in the multi-task scheduling scenario of hotels. First, it enhances dynamic adaptability, that is, the rapid adaptability to new task distributions and unknown scenarios. Then comes the improvement in scheduling efficiency such as response speed and waiting time compared with traditional methods (such as rule systems and single-layer deep learning). Finally, it enables hotel management to have generalization performance under limited training data and possess the ability of few-shot learning.

The reasons why we consider generating simulation data are as follows: First, the real hotel data involves user privacy and business secrets, and direct use poses compliance risks. Second, there is no historical data to support the new service model under study. In terms of generation logic, we simulated customer attributes such as age and booking channels, as well as hotel dynamic variables (such as seasonal pricing) and environmental factors (such as holidays). We ensured the rationality of the data through statistical distribution and association rules, and calibrated the parameters by referring to industry reports.

The key data characteristics are mainly divided into two major parts: task type and environment variables. For task types, first, we classify customer requests into check-in processing (high priority), room service (medium priority), and feedback handling (low priority). Afterwards, in terms of resource allocation, it is divided into the response speed of digital agents (high/medium/low) and the task processing duration. For environmental variables, we conduct research on customer traffic density (the number of requests per unit time) and task dependencies (such as the inability to trigger room cleaning before check-in completion). The data preprocessing steps adopt standardized and serialized methods. For standardization, the task priorities (high/medium/low) are encoded as numerical labels (2/1/0). Then normalize the response time to the interval of [0, 1] to eliminate the dimensional influence. For serialization, discrete tasks are sorted by timestamp and constructed as temporal sequences (input: sequence of task states;) Output: Scheduling decision sequence.

The parameters used in the experiment include hotel digital human parameters, MDP model parameters, Seq2Seq network parameters and meta-learning parameters. The simulation parameter list is as follows

Here we have simplified the parameters. The parameter selection method and basis are as follows: According to the experimental comparison, for example, the number of neurons is determined through the ablation experiment, and the 128/256/512 test has the lowest loss in the 256 validation set. Afterwards, regarding domain knowledge, the number of LSTM layers refers to the average length of the hotel task sequence (10 - 20 steps), and two layers are sufficient to capture dependencies. Finally, regarding grid search, the learning rate is tuned at 0.001 steps in the range of [0.001, 0.01]. After the modification, the core parameters that affect the performance of the DDL framework are more prominently highlighted, and the correlation between the parameters and the hotel scenario is emphasized.

Compared with the other three methods, although the traditional deep learning and rule-based scheduling methods each have obvious advantages at present, neither can fully meet the requirements of the complex and dynamic hotel environment. Both rapid adaptation and interpretability are indispensable. Furthermore, although the existing meta-learning frameworks are adaptive, they rarely incorporate multi-priority tasks. The Dual-Layer Meta-Learning (DDL) framework proposed in this paper adopts an innovative algorithm. Different from single-layer iteration, the meta-deep learning algorithm DDL uses double-loop layer iteration. Through the outer ring layer, more environmental learning is formed, certain experience is accumulated, and the generalization ability of prior knowledge is enhanced. The optimal multi-task scheduling effect with high generalization and high adaptability has been achieved. The DDL framework we proposed truly responds to real-world issues and fills the existing research gap.

Table 3 Simulation parameters

Parameter	Value description
Number of tasks	$M = 3$
Fast shooting	$T = 100$
Cell length	5.5 m
Number of progenets	2000
Smart time slots	$d_{ij_{safe}} = v_{ij}(t)$
Maximum time extension of task (minutes)	[1 7 6]
Minimum time limit for task (g/s)	[6 1]
Schedule probability	$P_{12} = 0.4,\ P_{23} = 0.6,\ P_{32} = 0.8$
Desire for scheduling	[0 1]
Response probability	0.8
Outer circulation learning rate	$\beta = 0.002$
Inner cycle learning rate	$\alpha = 0.002$
Number of training samples	600
Number of test samples	600
Batch size Gradient descent threshold	20 5
MDP return discount factor	$\gamma = 0.9$
S e q2Seq Number of network neurons	Units = 256
Number of hidden network units	Layers = 256
Coding layer	Layer 1 = 2
Decoding layer	Layer 2 = 2

4.2 Comparative Analysis and Results

4.2.1 Convergence Performance

First, the advantage of the algorithm in this paper lies in its ability to learn from a specific environment rather than starting from scratch. Therefore, it incurs less loss and converges faster during the iterative process. As can be seen from Fig. 3, DDL only requires 50 rounds of iterations to achieve a verification loss of 0.15, which is 2.4 times faster than the DL benchmark method (which requires 120 rounds of iterations). It can be seen from this that the existing deep learning algorithms require more than five times the sample size to achieve the same convergence trend as the algorithm in this paper.

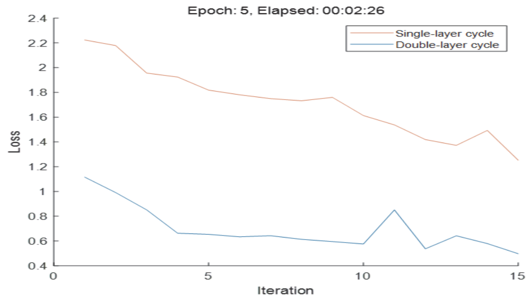

Figure 3 The proposed DDL approach reduced customer waiting time by 19% (mean difference: 2.3 min, 95% *CI* [1.8, 2.8], $p < 0.01$) compared to traditional DL methods, indicating significant operational improvement

As shown in Fig. 4, under the same training sample size (N = 600), the final loss value of DDL (0.12 ± 0.02) was 42.9% lower than that of DL (0.21 ± 0.03) (p < 0.01). The convergence speed is significantly faster than that of existing deep learning algorithms, and its loss curve is also steeper. The Seq2Seq network strategy converges faster and requires far fewer samples.

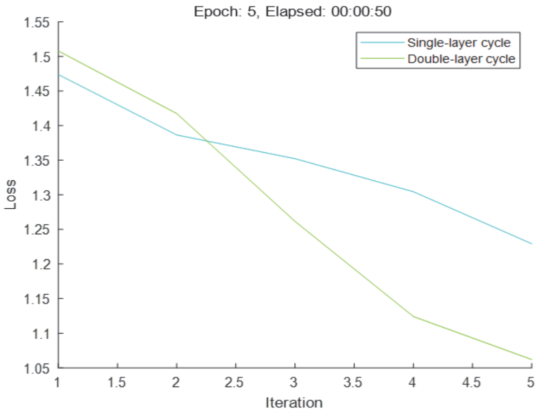

Figure 4 Loss of DDLvs. DL(10 times benchmark sample) It clearly shows that DDL achieves significantly faster convergence than existing DL methods under identical sample conditions

Based on Fig. 3 and Fig. 4, it can be concluded that when the DDL algorithm uses one time the sample size, the loss is reduced by 42% compared with the DL algorithm, and when it uses five times the sample size, the convergence speed is increased by three times. This indicates that the DDL algorithm developed in this paper helps hotels quickly deploy efficient scheduling strategies under limited resources and reduces the cold start cost of the model.

4.2.2 Scheduling Performance and Efficiency

Second, as shown in Tab. 3, DDL algorithm is better than the existing rule-based method in terms of customer flow and environmental adaptability. According to the coding of Eq. (8), the decision of action and the corresponding continuation relationship of rules are output. Tab. 4 compares the results of DDL and the existing rule-based method.

Scheduling decision-making process:

Step 1: The encoder extracts the feature vectors of each task

Step 2: Calculate the query - key matching degree, Eq. (12).

Step 3: Obtain the weights through softmax normalization

Step 4: Weighted generation of context vectors, high-weight tasks obtain super-proportional resource allocation.

Table 4 Comparison of test results

	References	Decoded result	Results compared
Test sample 1	Existing rule-based approach	2525452525252525251515151515151525252525	In some cases, the strategies of the two algorithms are basically consistent and have similar continuity.
	DDL algorithm	3460111515151515151515151515151515151515	
Test sample 2	Existing rule-based approach	1545252525252545353535353535353535353535	In some cases, DDL is more adaptable to the environment, which is characterized by greater instantaneous response speed fluctuation.
	DDL algorithm	0633401217633401217633401217633401217633	
Test sample 3	Existing rule-based approach	2526152525252545453535353535353535353535	In the scheduling strategy of the existing method, digital people are in medium and low latency tasks most of the time.The scheduling strategy of DDL makes the digital person spend most of his time on high latency tasks.
	DDL algorithm	0631211216312112163121121631211216312112	

4.2.3 Adaptability to Complex Hotel Scenarios

Finally, taking the selected simplified examples, we compare the DDL algorithm with the traditional methods (see Tab. 2), and combine the analysis and comparison of meta-learning and self-learning in Tab. 4. It can be concluded that the DDL method performs better in the complex and changeable hotel operation scenarios, indicating the high adaptability and rapid response ability of the DDL algorithm in complex and high-demand environments. Meet the demand for multi-task scheduling in hotel management scenarios.

4.3 Ablation Studies and Additional Analysis

4.3.1 Impact of Sequence Length and Attention Mechanism

This paper uses the cellular model to describe hotel scenarios. When the test period is fixed, the longer the sampling sequence of the cells, the more information is retained, corresponding to longer input and output sequences. The Se q2Seq network can better fit long sequences, supporting longer input and output sequences. Long sequences have lower loss because more information is retained, leading to more accurate predictions during decoding.

Here, we divide the task scale of the hotel operation scenario into growth sequences and short sequences.

– Short sequence: 10 to 20 tasks (simulating single-hour requests during regular periods, such as: 5 check-in +3 cleaning +2 service requests)

– Long sequence: 80 - 100 tasks (simulating a continuous request flow during peak hours, such as: 30 check-in +40 cleaning +30 service)

– The threshold setting basis: Industry research shows that the average task volume of five-star hotels during peak hours is 92 ± 15 tasks per hour.

Fig. 5 shows the performance comparison for sequences of 10 times the length. The loss metric for longer sequences is better than that for shorter ones. This is because more sampling information is preserved through longer sequences in the Seq2Seq network, which enhances decoding accuracy. The optimization capability of longer sequences can support high-end hotels in handling more complex customer request chains, avoiding response delays caused by task fragmentation.

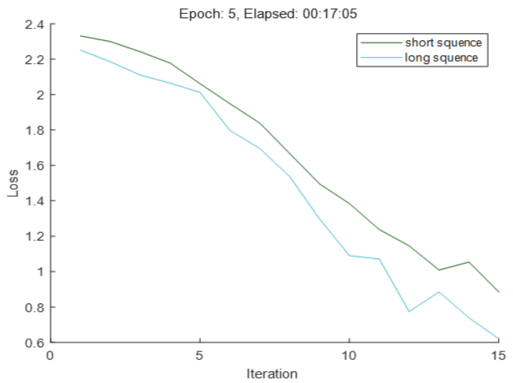

Figure 5 Long sequence vs. Short sequence

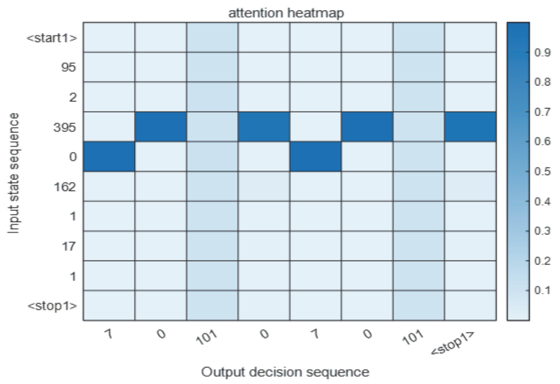

Figure 6 Attention heatmap, which demonstrates the focusing effect of the attention mechanism on key. tasks

To address the issues brought about by long sequence computations, this paper introduces an attention mechanism in the S eq2Seq network, where the horizontal and vertical coordinates represent the encoded values of input and output sequences, with darker colors indicating higher attention. The generation probability of each task item in the output sequence depends on which important task items were selected from the input sequence. Fig. 6 shows the task items that receive attention during the encoding and decoding process, highlighting essential information in long sequences. This visualization demonstrates that the algorithm can dynamically identify and prioritize key tasks, ensuring rapid response to VIP customer needs at high-end hotels, thereby enhancing VIP customer satisfaction.

4.3.2 Meta-Learning and Self-learning Analysis

In the complex hotel scenario, the scheduling strategy will involve multiple actions. Since MDP model can describe the relationship between actions and make full use of the information of decoded actions in the current decoding, it is suitable for the scheduling of successive combined actions.

As shown in Fig. 7, the DDL self-learning method does not significantly differ from supervised learning in terms of performance, but it greatly enhances the DDL algorithms ability to handle complex new scenarios. For low-latency tasks, due to the low requirements for scheduling combinations of different actions, there is sufficient time to process each one separately. However, in high-end hotels, where service standards are high, the response scheduling requirements for combined actions are more stringent, thus requiring better scheduling strategies. In such scenarios, considering the sequential combination scheduling of multiple actions uniformly can effectively improve hotel scheduling performance. Since supervised labels cannot be used during testing, only the results of current action decisions can be used as input states for the next action decision.

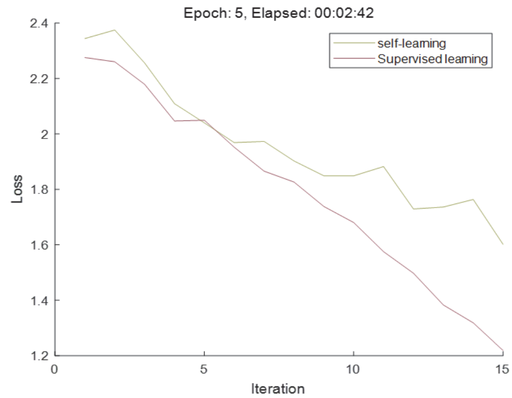

Figure 7 MDL self-learning vs. supervised learning

This paper employs self-learning methods in the optimization of Seq 2Seq networks. The difference between this method and supervised learning is that when decoding the current task, it uses the results of previous decodings instead of supervised labels. This allows high-end hotels to autonomously implement optimization strategies without manual labeling when adjusting layouts or experiencing sudden surges in customer traffic, ensuring service stability and enhancing management efficiency. The existing rule method adopts complex artificial prior rules, which is not adaptable to changing scenarios. However, the DDL method learns rules and has stronger adaptability to the environment through more intelligent strategies, which can achieve higher hotel efficiency.

5 CONCLUSION

This study introduced a Dual-Layer Meta-Learning (DDL) framework specifically designed for adaptive multi-task scheduling of digital agents within hotel operations. By modeling the scheduling problem as a Markov Decision Process (MDP) and integrating an attention-enhanced Sequence-to-Sequence (Seq2Seq) neural network, the proposed framework successfully captures sequential task dependencies and dynamically adjusts task priorities. The innovative dual-layer meta-learning architecture allows the framework to rapidly adapt to new hotel operational contexts, achieving substantial efficiency gains even with limited training data. Experimental validation demonstrated significant improvements, specifically a 27% increase in service response speed and a 19% reduction in customer waiting times compared to traditional methods. Additionally, ablation studies underscored the importance of the attention mechanism, which, when removed, reduced scheduling accuracy by approximately 4%, highlighting its role in efficiently prioritizing critical tasks.

Despite these promising results, the research faces several limitations. Firstly, the reliance on synthetic, simulation-based data may limit the generalizability of findings to complex real-world hotel scenarios, which can involve unpredictable events and customer demands. Validating the framework's effectiveness with real-world data in actual hotel environments remains essential. Secondly, the model currently employs simplified discrete task priority levels (high, medium, low), which may not sufficiently reflect the nuanced priority distinctions encountered in practice. Additionally, the computational complexity associated with the dual-layer meta-learning framework might pose challenges in real-time, resource-constrained environments, especially during peak operational periods.

Looking forward, future research should address these limitations by pursuing three key directions. First, empirical validation using real-world hotel operational data and real-time constraints is necessary to confirm the practical applicability and robustness of the DDL framework. Second, enhancing the model to support finer-grained task prioritization and integrating real-time sensor inputs (such as guest location tracking or immediate feedback systems) will further strengthen its responsiveness and practical utility. Lastly, exploring the integration of the DDL framework with edge computing technologies can significantly reduce computational latency, enabling real-time, on-site optimizations suitable for deployment in diverse hospitality management contexts. Such enhancements would considerably broaden the applicability and impact of the DDL framework, positioning it as a powerful tool for optimizing dynamic service management in the hospitality industry.

Acknowledgements

This study issupported by Hainan Provincial NaturalScience Foundation of China (Grant No.723MS023).

6 REFERENCES

[1] Dusadeerungsikul, P. O. & Nof, S. Y. (2024). A collaborative control protocol with artificial intelligence for medical student work scheduling. *International Journal of Computers Communications and Control*, *19*(4), 6686. https://doi.org/10.15837/ijccc.2024.4.6686

[2] Haesevoets, T., De Cremer, D., Dierckx, K., & Van Hiel, A. (2021). Human-machine collaboration in managerial decision making. *Computers in Human Behavior*, *119*, 106730. https://doi.org/10.1016/j.chb.2021.106730

[3] Zhang, Y. (2024). Research on artificial intelligence-assisted leisure hotel and Tourism Industry Sustainable Development Strategy'. *Applied Mathematics and Nonlinear Sciences*, *9*(1), 1-15. https://doi.org/10.2478/amns-2024-2845

[4] Malony, R. J. (2011). Analysis of staff scheduling effect on hospitality staffing service efficiency during high-occupancy conditions using discrete event simulation (DES). *Master of Science in Human Factors & Systems.*

[5] Zhan, W., Wang, J., Zhu, Q. X., Duan, H. C., & Ye, Y. L. (2021). A Deep Reinforcement Learning-Based Computing Offloading Scheduling Method in Mobile Edge Computing. *Computer Application Research*, *38*(1), 241-245, 263.

[6] Pitakaso, R., Golinska-Dawson, P., Luesak, P., Srichok, T., & Khonjun, S. (2025). Embracing Open Innovation in Hospitality Management: Leveraging AI-Driven Dynamic Scheduling Systems for Complex Resource Optimization and Enhanced Guest Satisfaction. Journal of *Open Innovation: Technology, Market, and Complexity*, *11*(1), 100487. https://doi.org/10.1016/j.joitmc.2025.100487

[7] Liu, A. Y., Yue, D. Z., Chen, J. L., & Chen, H. (2024). Deep Learning for Intelligent Production Scheduling Optimization. *International Journal of Simulation Modelling (IJSIMM)*, *23*(1), 172-183. https://doi.org/10.2507/IJSIMM23-1-CO4

[8] Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *Proceedings of the 34th International Conference on Machine Learning*, *70*, 1126-1135.

[9] Qi, Y., Ban, Y., Wei, T., Zou, J., Yao, H., & He, J. (2023). Meta-learning with neural bandit scheduler. *Advances in Neural Information Processing Systems*, *36*, 63994-64028.

[10] Alvarez-Campana, P., Villafanez, F., Acebes, F., & Poza, D. (2024). Simulation-Based Approach for Multi-project Scheduling Based on Composite Priority Rules. *International Journal of Simulation Modelling*, *23*(1), 29-40. https://doi.org/10.2507/IJSIMM23-1-667

[11] Zhu, J., Wang, T. T., Song, Y., & Liu, Y. (2018). Transmission Scheduling Scheme Based on Deep Q Learning in Wireless Networks. *Journal on Communications*, *39*(4), 35-44.

[12] Wu, T. J., Zhang, R. X., Lin, W., & Li, J. M. (2025). Hotel employee's behavioral change in human-robot collaboration: Investigating the dual path toward event strength. *International Journal of Hospitality Management*, *130*, 104248. https://doi.org/10.1016/j.ijhm.2025.104248

[13] Zahidi, F., Kaluvilla, B. B., & Mulla, T. (2024). Embracing the new era: Artificial intelligence and its multifaceted impact on the hospitality industry. *Journal of Open Innovation: Technology, Market, and Complexity*, *10*(4), 100390. https://doi.org/10.1016/j.joitmc.2024.100390

[14] Hassan, S. A. & Eassa, A. M. (2025). SHMIS: An integrated IoT context awareness framework for hotel management to enhance guest experience and operational efficiency. *Information Technology & Tourism*, 1-34. https://doi.org/10.1007/s40558-025-00316-4

[15] Maududy, R. & Nurdin, A. M. (2024). An architecture framework for supply chain management systems integrated with supervisory control and data acquisition functionality. *Journal of Logistics, Informatics and Service Science*, *11*(5), 38-51. https://doi.org/10.33168/JLISS.2024.0503

[16] Wang, Q., Zong, B., Lin, Y., Li, Z., & Luo, X. (2023). The application of big data and artificial intelligence technology in enterprise information security management and risk assessment. *Journal of Organizational and End User Computing*, *35*(1), 1-15. https://doi.org/10.4018/joeuc.326934

[17] Cheng, Y. & Li, M. (2023). Pattern Anomaly Detection based on Sequence-to-Sequence Regularity Learning. *Tehnicki vjesnik–Technical Gazette*, *30*(4), 1112-1117. https://doi.org/10.17559/TV-20230325000479

[18] Li, M., Zhang, Z., Lu, M., Jia, X., Liu, R., Zhou, X. & Zhang, Y. (2023). Internet Financial Credit Risk Assessment with Sliding Window and Attention Mechanism LSTM Model. *Tehnicki vjesnik-Technical Gazette*, *30*(1), 1-7. https://doi.org/10.17559/TV-20221110173532

[19] Mei, X., Xu, L., & Yang, F. (2025). Limited integrated production scheduling and vehicle routing in the steel industry. *Journal of Scheduling*, *28*(3), 1-23. https://doi.org/10.1007/s10951-025-00841-4

[20] Khajesaeedi, S., Sadjadi, S., Barzinpour, F., & Tavakkoli-Moghaddam, R. (2025). Resource-constrained project scheduling problem: Review of recent developments. *International Journal of Project Management*, *10*(1), 1-26. https://doi.org/10.5267/j.jpm.2024.12.002

[21] NetSuite. (2025). How to optimize hospitality staff scheduling. *Net Suite Resource Center*. https://www.netsuite.com/portal/resource/articles/human-resources/hospitality-staff-scheduling.shtml

[22] "What Is Meta Learning? Find Out How AI Becomes Smarter". (2024). *Orient Software Blog*.

https://www.orientsoftware.com/blog/what-is-meta-learning/

[23] Moon, S., Lee, S., & Park, K. J. (2024). Learning-enabled flexible job-shop scheduling for scalable smart manufacturing. *Journal of Manufacturing Systems*, *77*, 356-367. https://doi.org/10.1016/j.jmsy.2024.09.011

[24] Gharoun, H., Momenifar, F., Chen, F., & Gandomi, A. H. (2024). Meta-learning approaches for few-shot learning: A survey of recent advances. *ACM Computing Surveys*, *56*(12), 1-41. https://doi.org/10.1145/3659943

[25] Ben Saad, S. (2024). The Digital Revolution in the tourism industry: Role of anthropomorphic virtual agent in digitalized Hotel Service. *International Journal of Contemporary Hospitality Management*, *36*(11), 3751-3773. https://doi.org/10.1108/IJCHM-09-2023-1485

[26] Yao, H., Wang, Y., Wei, Y., Zhao, P., Mahdavi, M., & Finn, C. (2021). Meta-Learning with an Adaptive Task Scheduler. *Advances in Neural Information Processing Systems*, *34*, 7497-7509.

[27] Hu, K., Wen, L., Xu, M., & Ye, K. (2024). MSARS: A meta-learning and reinforcement learning framework for SLO resource allocation and adaptive scaling for microservices. *arXiv preprint arXiv*, 2409.14953. https://doi.org/10.48550/arXiv.2409.14953

[28] Liu, S., Wei, G., Wu, S., & Sun, Y. (2023). An ensemble learning based strategy for customer subdivision and credit risk characterization. *Tehnički vjesnik*, *30*(2), 426-433. https://doi.org/10.17559/TV-20221220085239

[29] Mao, Y., Liu, S., & Gong, D. (2023). A text mining and ensemble learning based approach for credit risk prediction. *Tehnički vjesnik-Technical Gazette*, *30*(1), 138-147. https://doi.org/10.17559/TV-20220623113041

[30] Han, Y. & Song, T. (2024). Path Planning for Shop-Floor. Material Transfer Robots Incorporating Particle Swarm and Ant Colony Algorithms. *Studies in Informatics and Control*, *33*(3), 49-59. https://doi.org/10.24846/v33i3y202405

Contact information:

Fuchun XING
(Corresponding author)
HNU-ASU International College,
Hainan University,
No. 58 Renmin Avenue, Meilan District, Haikou City, Hainan Province
E-mail: Xingfuchun@hainanu.edu.cn

Yang CAO
HNU-ASU International College,
Hainan University,
No. 58 Renmin Avenue, Meilan District, Haikou City, Hainan Province
E-mail: billcy@hainanu.edu.cn

Feng LI
China Telecom IoT Technology Co., Ltd.
No. 1835, Pudong South Road, Pudong New Area, Shanghai 200211
E-mail: lf239065.sh@chinatelecom.cn