Transactions on Computational and Scientific Methods | Vo. 4, No. 05, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Global Context Modeling and Structure-Guided Feature Enhancement for Queue Time Prediction in Large-Scale Cloud Systems

Hong Zhuang[1], Kevin Gao[2]
[1]University of Illinois at Urbana Champagne, Champagne, USA
[2]University of South Carolina, Columbia, USA
*Corresponding author: Hong Zhuang; hongz3@illinois.edu

Abstract: This paper proposes a waiting time prediction framework that integrates global semantic modeling with structure-aware enhancement, aiming to improve the accuracy and stability of task response time in complex scheduling systems. A Global Context-Aware Regression Module is designed to model deep semantic dependencies between tasks and system states, enhancing the model's understanding of scheduling behavior. In addition, a Structure-Guided Feature Enhancement mechanism is introduced, using scheduling structures and task-phase information as priors to perform residual modeling and semantic alignment of intermediate features. This improves representation consistency and discriminative ability under task concurrency and resource fluctuations. Experiments are conducted on a public scheduling dataset, and multiple evaluation metrics are used to compare prediction performance across models, demonstrating the proposed method's advantages in accuracy, robustness, and structural generalization. Further ablation studies and hyperparameter sensitivity analysis confirm the collaborative effect of each module in maintaining model stability and representation capability, validating the effectiveness and applicability of the overall framework.

Keywords: Queue time modeling; scheduling system prediction; feature alignment mechanism; residual information modeling

1. Introduction

Deep In scheduling systems where task concurrency and resource heterogeneity coexist, waiting time prediction is becoming a critical task for ensuring system responsiveness and resource efficiency. As complex systems such as intelligent manufacturing, cloud platforms, and high-performance computing continue to grow, the interactions between task requests and system behaviors become increasingly complex. Traditional prediction methods that rely on rules or shallow features show a significant decline in adaptability under highly dynamic conditions. These methods fail to meet the practical demand for high-precision prediction. Building a waiting time prediction model with strong robustness and generalization has become a key challenge for advancing intelligent scheduling [1, 2].

Although some studies have introduced deep learning models to enhance the modeling of tasks and system states, many challenges remain. First, existing methods are weak in capturing semantic relationships among tasks and context-aware scheduling dependencies, leading to incomplete feature representations. Second, they often ignore structural constraints and task-phase information in the scheduling process. As a result, they cannot accurately model behavior evolution under resource competition [3]. Third, these methods lack effective

mechanisms for handling dynamic system uncertainty and feature alignment errors. This leads to poor generalization and limited model stability.

To address these challenges, this paper proposes a waiting time prediction framework that integrates context-awareness and structure-guided mechanisms. The framework combines global semantic modeling and local structural enhancement to build a unified task-system representation space. It improves the model's ability to capture scheduling behaviors. The overall method consists of a context enhancement module, a structure-guided module, and a supervised regression module. It shows good portability and adaptability across different application scenarios [4].

The main contributions of this study are as follows. (1) A Global Context-Aware Regression Module (GCARM) is introduced to capture deep semantic dependencies between tasks and system states. This enhances the model's capacity to represent complex scheduling relationships. (2) A Structure-Guided Feature Enhancement mechanism (SGFE) is designed to incorporate task-phase information and structural priors from the scheduling process. It guides residual feature modeling and feature alignment, improving the stability of feature representations. This study provides a transferable modeling paradigm for building high-performance waiting time prediction models in heterogeneous scheduling systems [5, 6].

2. Related work

2.1 Task Waiting Time Modeling in Cloud Computing Systems

In large-scale cloud computing systems, task waiting time is one of the key performance metrics. It is widely used to evaluate resource scheduling efficiency and system responsiveness. Traditional schedulers often rely on static rules or fixed-priority strategies to assign tasks. They estimate the expected waiting time based on task submission order, resource request volume, or estimated execution duration. However, these approaches usually ignore dynamic changes during system operation and multi-tenant interference [7]. As a result, they struggle to accurately capture task queuing behavior under high concurrency. In particular, under heavy resource contention, task waiting time becomes highly volatile and unpredictable. This greatly limits the applicability of static models [8].

To address the issues of high prediction bias and low adaptability in traditional methods, some studies have attempted to introduce queuing theory to model the task waiting process. These approaches often assume that the resource service process follows certain probability distributions, such as Poisson arrivals and exponential service times. They use these assumptions to derive analytical expressions for average waiting time prediction [9]. However, in real-world cloud platforms, task arrivals are influenced by user behavior, business patterns, and resource contention. These factors result in non-stationary characteristics, making classical queuing models inadequate in accuracy. Additionally, task heterogeneity, such as varying execution durations and different resource requirements, makes it difficult to satisfy the assumptions of traditional queueing models [10].

With the advancement of virtualization and container technologies, the scheduling granularity in cloud systems has become increasingly fine-grained. The queuing behavior is also becoming more complex. There exists a highly coupled dynamic relationship among task execution order, scheduling latency, and system load. Waiting time is influenced by the joint effects of many factors. In multi-tenant environments, resource preemption, scheduling conflicts, and system-level interference co-exist. These challenges make waiting time prediction more difficult. Furthermore, the scheduler must consider both current system status and historical behavior in each decision round. Modeling from a single perspective can no longer handle such highly dynamic and concurrent systems [11].

Against this backdrop, how to model task waiting time more comprehensively and precisely has become a major research focus in cloud computing systems. In real-world operational environments, empirical observations consistently indicate that waiting time is influenced not only by intrinsic properties of the submitted tasks—such as priority, type, and resource demands—but also by a broader set of contextual factors. These include real-time system resource utilization levels, the evolution of scheduling strategies over time, the

cumulative impact of historical scheduling behaviors, and the unpredictable fluctuations in node-level workloads. Collectively, these elements exhibit complex characteristics such as high nonlinearity, intricate long-term temporal dependencies, and strong multi-timescale correlations, all of which significantly elevate the difficulty of accurate modeling. Consequently, estimation approaches that rely solely on static task descriptors or rule-based heuristics are fundamentally limited in their ability to capture the dynamic behaviors and structural variations inherent in modern cloud infrastructures. To address these limitations, there is an urgent need for expressive and adaptive modeling paradigms capable of learning fine-grained temporal patterns and contextual interactions to support high-fidelity prediction of task waiting behaviors under diverse and evolving system conditions [12].

2.2 Learning-Based Approaches for Resource Scheduling and Time Prediction

Facing the limitations of traditional queuing time modeling in capturing dynamic and nonlinear features, recent studies have increasingly explored data-driven learning methods to improve the prediction of scheduling-related time metrics. These methods rely on historical task execution logs, system load information, and scheduling behavior sequences. By training learnable models, they aim to predict task waiting time, execution time, or resource usage. In cloud computing systems, such models can uncover hidden patterns from large-scale scheduling data [13]. They can also perform adaptive modeling in highly heterogeneous and time-varying environments, helping schedulers make more efficient resource allocation decisions.

Early learning-based approaches mostly adopted supervised regression frameworks, formulating task waiting time prediction as a feature-to-value mapping problem [14]. These methods typically construct input vectors from static task features, such as submission time, requested resources, or historical average waiting time. They applied classical machine learning models like decision trees, support vector machines, or random forests. While these approaches worked reasonably well in small-scale and relatively stable environments, they struggled in large cloud platforms with highly imbalanced task distributions and rapidly changing system states. Their limited capacity in feature representation and context awareness made it difficult to capture deep interactions between task behavior and system dynamics [15].

With the progress of deep learning techniques, researchers began to apply neural network architectures to task time modeling and scheduling prediction. Recurrent neural networks such as LSTM and GRU, along with attention-based models, were introduced to better handle temporal dependencies and scheduling pattern evolution across different stages [16]. Some works further adopted graph neural networks to model task dependencies and the structure of scheduling graphs. These methods improved the model's ability to represent complex structures in multi-task scheduling scenarios. In addition, several studies integrated multi-source input signals, including resource usage streams, scheduling history trajectories, and load fluctuation trends. These enriched features enhanced the model's understanding of contextual information [17].

Although learning-based methods show stronger flexibility and higher prediction accuracy, they still face several inherent challenges in practical scheduling scenarios. On one hand, the large-scale scheduling logs collected from complex systems often exhibit characteristics such as noise contamination, structural diversity, and incomplete records. These issues can significantly impair the stability of the training process and hinder the generalization ability of the model when applied to unseen scheduling conditions. On the other hand, despite the powerful nonlinear modeling capacity of deep learning architectures, they tend to operate as black boxes, lacking transparency in their internal decision-making mechanisms. This absence of interpretability poses serious obstacles in real-world deployments, as system administrators and operators often require explainable predictions to ensure safety, compliance, and operational reliability. In addition, scheduling tasks in cloud and high-performance computing environments is usually constrained by strict latency and throughput requirements, which demand not only high prediction accuracy but also lightweight model structures capable of real-time inference and continuous online adaptation. As a result, it remains an open and essential challenge to develop context-aware, structurally robust, and computationally efficient learning frameworks tailored to dynamic task time modeling in heterogeneous scheduling systems [18, 19].

3. Method

To address the challenges of multi-source heterogeneous feature fusion and insufficient dynamic context awareness in task waiting time modeling for large-scale cloud computing systems, this paper proposes a dual-branch regression framework. The method introduces two core innovations. First, a Global Context-Aware Regression Module (GCARM) is designed to capture cross-time and cross-node associations between tasks and system states, enhancing the model's ability to represent the evolution of scheduling behaviors. Second, a Structure-Guided Feature Enhancement (SGFE) mechanism is developed to integrate scheduling queue structures, resource topology states, and task concurrency patterns. This enhances the model's sensitivity and robustness to waiting time fluctuations in highly dynamic environments. The overall framework demonstrates strong expressiveness and adaptability, enabling accurate modeling under varying system loads and resource distributions. The detailed structure of the proposed model is illustrated in Figure 1.

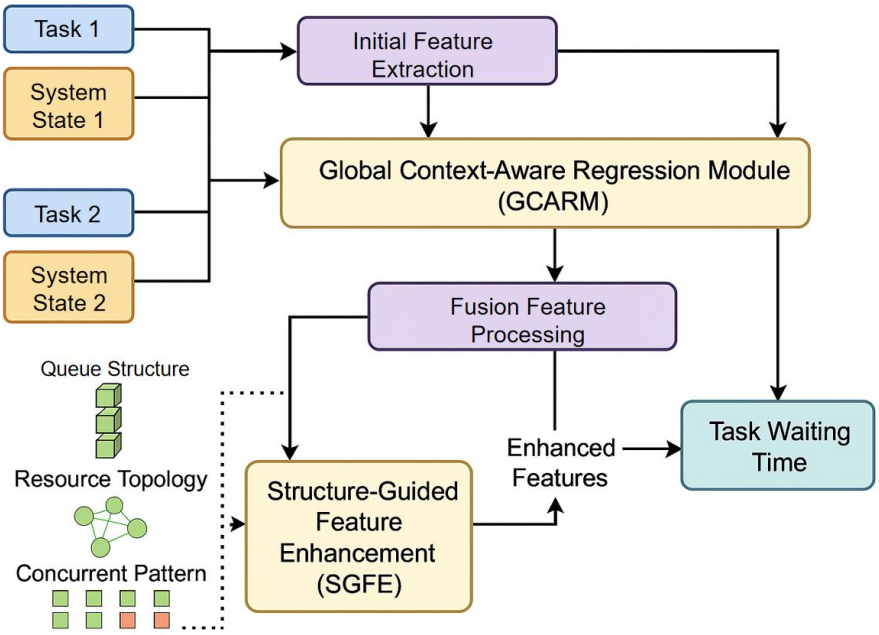

Figure 1. Overall model architecture diagram

3.1 Global Context-Aware Regression Module

This study introduces a Global Context-Aware Regression Module (GCARM) to deeply model the implicit global dependencies between task features and system states. The goal is to significantly enhance the ability to perceive scheduling context during waiting time modeling. The workflow of the module is shown in Figure 2. In large-scale cloud computing environments, task queuing behavior is influenced not only by task-specific attributes but also by overall system load, dynamic changes in scheduling strategies, and inter-task interference. Traditional methods often overlook these cross-task and cross-time structural dependencies, leading to poor responses to complex behaviors in real scheduling scenarios. Therefore, GCARM is designed to model task-environment relationships from a global system perspective, improving the completeness and consistency of feature representations.

The structure of GCARM consists of three key stages: context encoding, global interaction modeling, and regression supervision. First, the module encodes joint inputs of multiple tasks and their corresponding system states to extract a context vector that represents the global scheduling state. Then, this context information is injected into each task representation to achieve interactive fusion between task features and global system states. Finally, supervised learning is applied to the context-enhanced features for regression modeling of waiting time. The module produces more accurate and context-aware predictions. It also enhances the model's ability to adapt to complex dependencies and provides a stable and informative feature foundation for subsequent modules such as SGFE.

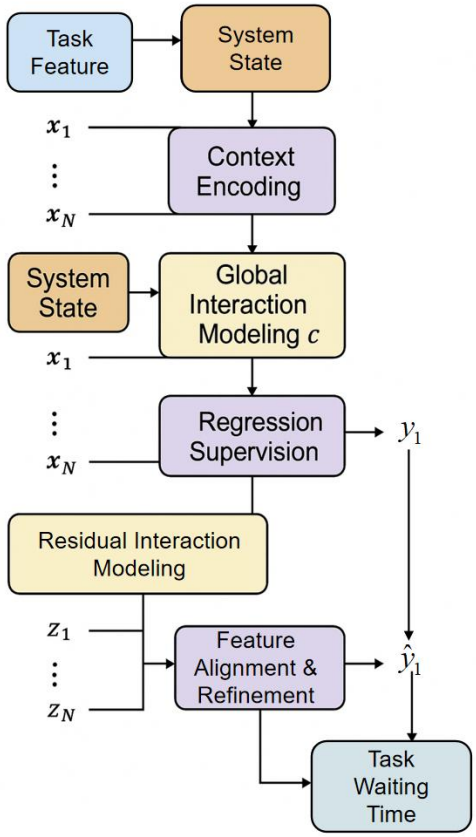

Figure 2. GCARM module architecture

Let the input be the joint feature vector of the i th task, denoted as $x_i \in R^d$, which contains multi-dimensional concatenation information from task attributes and system state. We first define a global context embedding function to encode the aggregate representation of all task inputs:

$$c = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)$$

Where $\phi(\cdot)$ represents a learnable feature transformation function (such as a multilayer perceptron or attention mechanism), $c \in R^h$ is the global context vector, and N is the number of input tasks. This context vector is used to guide context alignment modeling for each task.

To inject global context into the task feature modeling process, we construct a context-guided interaction function to fuse the context vector c with each task representation x_i to obtain the context-enhanced representation z_i:

$$z_i = x_i + \gamma \cdot \sigma(W_{1x_i} + W_{2c} + b)$$

Where $\gamma \in R$ is the learnable fusion weight, $\sigma(\cdot)$ is the nonlinear activation function (such as ReLU or GELU), W_1 and $W_2 \in R^{h\times d}$ are the transformation matrices, and $b \in R^h$ is the bias term. This fusion mechanism explicitly models the dependency between tasks and the global state.

After context fusion, the GCARM module uses the regression head to predict the context-enhanced representation z_i and generate the queue time estimate $\hat{y}_i$:

$$\hat{y}_i = f_{reg}(z_i) = w^T z_i + b$$

Where $f_{reg}(\cdot)$ is a linear regression function, and $w \in R^h$ and $b \in R$ are learnable parameters. This function is concise and easy to train with enhanced features to improve model stability.

To improve the model's accuracy in regressing context-aware information, we designed an objective function consisting of a main loss and a regularization term. The main loss term uses the standard mean squared error (MSE) to constrain the difference between the predicted value and the actual queue time:

$$L_{mse} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

Where y_i is the actual queue time label of the i th task. To enhance the generalization ability of the model and suppress the interference of unstable task characteristics on the prediction results, we introduce the L2 regularization term for structural constraints:

$$L_{total} = L_{mse} + \lambda(\| w \|_2^2 + \| W_1 \|_F^2 + \| W_2 \|_F^2)$$

Where λ is the regularization weight hyperparameter, $\| \cdot \|_2$ and $\| \cdot \|_F$ represent the L2 norm and Frobenius norm of the vector and matrix, respectively. This objective function not only optimizes regression accuracy but also enhances the robustness and stability of the model structure.

Finally, the loss function plays a critical role in optimizing the GCARM module. It imposes supervision on both regression accuracy and context-aware modeling capability. This encourages the model to learn stable and generalizable representations of task waiting time under dynamic scheduling environments. The loss function also provides global consistency support for the entire prediction framework.

3.2 Structure-Guided Feature Enhancement

This study introduces a Structure-Guided Feature Enhancement (SGFE) mechanism. It is designed to perform structure-aware modeling and fine-grained feature alignment on the intermediate representations generated during task-system interaction. The goal is to improve the model's discriminative power and generalization ability in task waiting time prediction. The workflow of this module is shown in Figure 3. In large-scale cloud computing environments, task queuing behavior is often influenced by complex factors such as scheduling queue structures, resource topology states, and concurrent execution patterns. These structural factors are often overlooked in traditional models, leading to a lack of responsiveness to system structural changes. By incorporating structural prior knowledge, SGFE provides more comprehensive context awareness, helping the model capture hidden dependencies and conflict patterns among tasks.

Specifically, this module uses structural information from the scheduling process as a guiding signal. It combines this with context-enhanced features from previous stages to perform residual modeling and semantic alignment. The residual modeling component reveals local behavioral deviations of tasks under structural interference. The semantic alignment mechanism unifies the feature representation scale to improve consistency among different tasks in the feature space. This mechanism not only preserves the global interaction features learned in earlier stages but also enhances feature accuracy and stability by incorporating structural details. As a result, the model can produce more robust waiting time predictions when facing dynamic scheduling scenarios with task competition and resource fluctuations.

Let the context-enhanced feature representation obtained from the upstream module be $z_i \in R^d$. The structure-guided enhancement mechanism first constructs a residual representation $r_i \in R^d$ based on the system scheduling structure. This representation is used to model the local perturbation deviation of the task. We define the following residual information modeling function:

$$r_i = \psi(z_i, s_i) = \sigma(W_z z_i + W_s s_i + b)$$

Here, $s_i \in R^d$ represents the structural representation extracted from the queue structure, resource topology, and concurrency graph, $\sigma(\cdot)$ is the nonlinear activation function, and W_z , $W_s \in R^{d \times d}$, and $b \in R^d$ are learnable parameters. This representation incorporates structural context and models subtle differences between tasks.

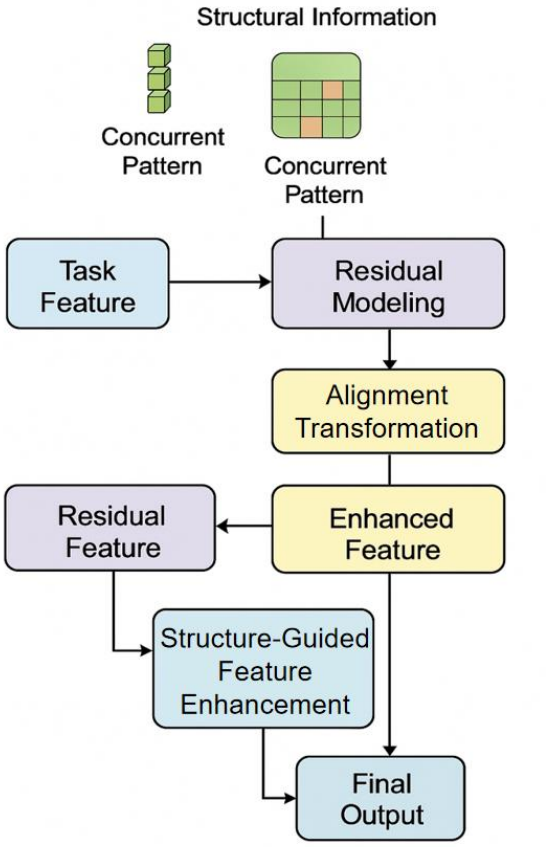

Figure 3. SGFE module architecture

Subsequently, the SGFE module fuses the residual representation with the original features to construct a structure-enhanced feature representation $\widetilde{z}_i$ for alignment and refinement:

$$\widetilde{z}_i = z_i + \alpha \cdot r_i$$

Where $\alpha \in R$ is a trainable scalar that controls the strength of the residual information on the main features. To further unify the semantic scales between different tasks, the module introduces a linear alignment transformation to obtain the final regression input vector h_i:

$$h_i = W_a \widetilde{z}_i + b_a$$

Among them, $W_a \in R^{d' \times d}$, $b_a \in R^{d'}$, the alignment process maps all features to a unified dimension d', which is convenient for subsequent regression supervision.

After obtaining the structurally aligned feature h_i, SGFE uses an independent prediction head to perform the final queue time estimation, and the output is $\hat{y}_i^{sgfe}$:

$$\hat{y}_i^{sgfe} = f_{reg}(h_i) = w^T h_i + b$$

To improve the discriminative ability of the structure enhancement module, we designed a joint optimization objective function, including the main regression loss and the structure alignment regularization term, defined as follows:

$$L_{sgfe} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i^{sgfe} - y_i)^2 + \beta \cdot \| r_i \|_2^2$$

The first term is the standard mean squared error loss, the second term is the residual regularization term, and β is a weight coefficient that controls the complexity of structural deviation modeling. This loss function not only optimizes prediction accuracy but also suppresses the interference of invalid structural perturbations on feature enhancement. This enhances the model's sensitivity to structural characteristics while improving prediction stability and consistency. Through this mechanism, the model establishes structural constraint associations between tasks, effectively improving its ability to model dynamic queue behavior.

4. Experimental Results

4.1 Dataset

This study uses the Cloud Workload Dataset for Scheduling Analysis as the main experimental data source. The dataset records real task scheduling traces from large-scale cloud computing systems. It includes key fields such as task submission time, resource requests, task duration, priority level, and execution node information. These attributes provide a comprehensive view of the context in which task waiting time is formed. Each task sample is annotated with its actual waiting time after submission, making it suitable for building regression models for waiting time prediction. This information is especially important for constructing context-aware task representations and provides rich input features for the GCARM module.

In addition to basic task-level attributes, the dataset also includes system-level features related to scheduling structures. These features cover resource topologies, concurrent task distribution graphs, and hierarchical queue information. Such structural data is represented in the form of graphs or sparse matrices. It reflects internal scheduling hierarchies and resource competition patterns in multi-tenant systems. These structural dimensions allow the extraction of inter-task interference and shared resource paths. They offer explicit guidance for structure-aware modeling in the SGFE module, enhancing the model's ability to represent complex scheduling environments.

Overall, the Cloud Workload Dataset for Scheduling Analysis provides a multimodal foundation for task waiting time prediction. It includes both static task attributes and dynamic system states with structural semantics. The dataset supports global context modeling while also enabling the extraction of structural dependencies and residual features. It is well-suited for building multi-stage scheduling learning frameworks and offers a unified and continuous database for joint optimization of the two modules.

4.2 Experimental Setup

The experiments in this study were conducted on a local server equipped with high-performance computing capabilities. The environment included the Ubuntu operating system and a CUDA-based GPU acceleration framework. The computing platform was configured with an NVIDIA RTX 3090 GPU (24 GB memory), an Intel Core i9-12900K CPU, and 128 GB DDR5 RAM. It supported multi-process data loading and mixed-precision training, meeting the computational needs of large-scale deep regression models. All models were implemented and executed using the PyTorch deep learning framework to ensure reproducibility and efficiency.

For optimization, all models used the Adam algorithm for parameter updates. The initial learning rate was set to 1×10^{-4}, and a fixed learning rate schedule was applied to maintain training stability. The batch size was set to 64. Validation was performed after each epoch, and both loss and regression performance metrics were recorded. To prevent gradient explosion, gradient clipping was applied with a maximum norm of 5. Weight decay was also used to improve model generalization, with the decay coefficient set to 1×10^{-5}.

The training process was run for a total of 300 epochs. An early stopping mechanism was activated starting from the 50th epoch to monitor changes in validation performance. To accelerate convergence, all input features were standardized before training. To support multi-scale structural inputs, automatic mixed precision (AMP) was used during training to reduce memory usage and improve computational efficiency. All experiments were conducted using a single GPU, ensuring that the setup satisfied reproducible and high-performance training requirements.

4.3 Experimental Results

4.3.1 Comparative Experimental Results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table 1. Comparative experimental results

Model	MAE ↓	RMSE ↓	MSLE ↓	R^2 ↑
esDNN [19]	12.4	18.6	0.014	0.81
MAG-D [20]	11.7	17.9	0.013	0.83
WGAN- gp Transformer [21]	10.9	16.8	0.012	0.85
Bayesian LSTM [22]	10.3	16.2	0.011	0.86
Ours	9.80	15.4	0.009	0.88

From the overall experimental results, the proposed model achieves superior performance across multiple evaluation metrics, demonstrating its effectiveness in modeling task waiting time under complex cloud computing environments. Traditional DNN-based models, such as esDNN, rely mainly on static task attributes for prediction. While they can capture local patterns, they struggle with nonlinear dependencies under dynamic scheduling scenarios. This results in higher overall errors, with noticeable fluctuations in RMSE and MSLE, indicating limited ability to fit sudden changes in waiting time behavior.

Models that incorporate temporal modeling mechanisms, such as MAG-D and WGAN-gp Transformer, show performance improvements. These models are more effective in capturing multivariable dependencies and task dynamics. MAG-D uses attention mechanisms and a bidirectional GRU to construct context-aware temporal representations. WGAN-gp Transformer further enhances the model's ability to fit data distributions through adversarial training, significantly reducing prediction variance. However, both models cannot model structural dependencies, such as task correlations arising from resource conflicts and queue topology, and thus cannot fully capture non-independent task behaviors.

The Bayesian LSTM model improves generalization by introducing uncertainty modeling. It performs well on RMSE and R^2, showing that explicitly modeling prediction intervals offers advantages in dynamic resource environments. However, its weak sensitivity to structural signals limits its stability in high-load and rapidly changing cloud platforms. Specifically, it still suffers from error accumulation at a fine-grained level, as seen in the MSLE metric.

Compared to the above methods, the proposed model achieves the best results across all evaluation metrics. This indicates that the joint modeling framework built with GCARM and SGFE offers stronger context modeling and structural adaptation capabilities. GCARM models temporal dependencies between tasks and system states by introducing global context modeling. SGFE enhances semantic alignment and residual feature representation by incorporating structure-guided modeling. Together, these modules significantly improve the model's ability to handle complex scheduling structures and prediction accuracy. This modular and structure-sensitive modeling approach enables the model to produce stable and accurate waiting time estimates even under high concurrency and resource contention. It highlights the feasibility and practicality of the proposed method for deployment in real-world cloud computing systems.

4.3.2 Ablation Experiment Results

To comprehensively evaluate the effectiveness of each component in the proposed model, ablation studies were further conducted. This experiment removes or replaces key modules step by step to analyze their impact on overall performance. The goal is to reveal the actual contribution of each structure during the modeling process. Such analysis helps validate the rationality of the model design and the synergy among different modules. Table 2 presents the performance changes after removing or replacing each core module based on the full model. By comparing results under different configurations, one can observe the role each component plays in improving prediction accuracy and model stability. This table provides empirical evidence for the following analysis of module effectiveness.

Table 2. Ablation experiment results

Method	MAE ↓	RMSE ↓	MSLE ↓	R^2 ↑
Baseline	12.7	18.9	0.015	0.81
+ GCARM	11.1	17.1	0.012	0.85
+ SGFE	10.6	16.6	0.011	0.86
+ All	9.80	15.4	0.009	0.88

According to the experimental results, the baseline model shows limited performance when modeling task waiting time using only local task attributes without global awareness or structural alignment. This modeling approach fails to capture the non-independent relationships among tasks and the dynamic evolution of system states. As a result, the model exhibits large prediction fluctuations under high concurrency or intense resource competition. It shows low stability and weak discriminative ability. This observation indirectly confirms that queuing behavior is a dynamic process influenced by multiple coupled factors. Modeling based on single-dimensional information is insufficient to meet the accuracy requirements of complex scheduling environments.

After introducing the Global Context-Aware Regression Module (GCARM), the model significantly improves its ability to capture interactions between system states and task histories. This module constructs context dependencies across tasks and across time, allowing the model to incorporate richer scheduling information during feature encoding. As a result, it enables more accurate waiting time estimation. Importantly, this mechanism does not simply aggregate information. Instead, it guides the dynamic evolution of the feature space through context fusion. This improves the model's adaptability to scheduling patterns, especially in scenarios with high task heterogeneity or large workload fluctuations.

On this basis, the addition of the Structure-Guided Feature Enhancement (SGFE) mechanism further enhances the model's understanding of structural information in scheduling. By incorporating variables such as queue structures, resource topologies, and concurrency patterns, the model can explicitly represent resource dependencies and scheduling interference among tasks. This allows for fine-grained feature alignment and residual compensation at the representation level. Structure-aware modeling improves feature consistency and strengthens the model's robustness in response to structural changes in the system. When the two modules work together, they complement each other in context modeling and structural modeling. This results in more stable and generalizable predictions of task waiting time.

4.3.3 Impact of Changes in Regularization Strength on Model Stability

This paper further analyzes the impact of changes in regularization strength on model stability, recognizing that the degree of regularization directly influences the balance between overfitting and underfitting during the training process. By systematically adjusting the strength of the regularization term, the study examines how different levels of constraint affect the model’s capacity to maintain consistent performance under varying conditions, particularly in scenarios where robustness and generalization are critical. This analysis provides important insights into how regularization functions not only as a mechanism for controlling complexity but also as a stabilizing factor that shapes the overall reliability of the predictive framework, with the corresponding experimental setup and comparative findings illustrated in Figure 4.

The impact of regularization strength on model performance shows a clear nonlinear trend in the task waiting time regression setting. When the regularization coefficient is small, the model fails to control complexity effectively, which increases the risk of overfitting. As a result, large prediction errors occur under scheduling fluctuations such as high concurrency and task heterogeneity. In particular, RMSE and MAE increase significantly when the regularization weight approaches zero. The model becomes overly sensitive to abnormal queuing behavior, reducing overall prediction stability. This indicates that models lacking proper regularization cannot handle distribution shifts and resource disturbances between tasks.

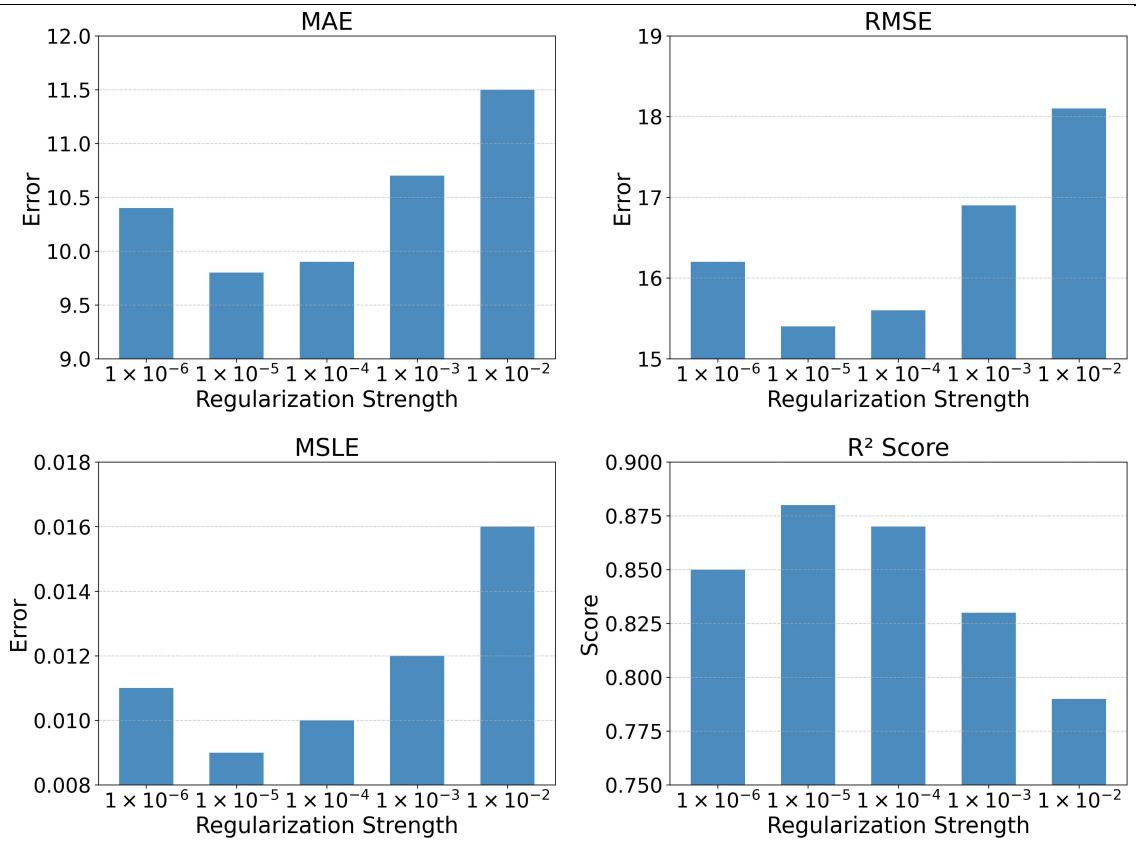

Figure 4. Impact of changes in regularization strength on model stability

As the regularization strength increases, the model demonstrates better generalization. Moderate weight decay improves the stability of residual information modeling and reduces redundancy across feature channels. This leads to more compact and organized enhanced features under the structure-guided mechanism. The improvement is especially notable in the MSLE metric, suggesting that the model maintains strong local discrimination when modeling both short and long delays. It also suppresses the influence of large values, making it more suitable for real-world scheduling systems with multi-scale queuing distributions.

However, when the regularization becomes too strong, the model's representation ability is restricted, resulting in a drop in overall performance. This suggests that excessive regularization weakens signal transmission in both the context interaction and structural residual pathways. It reduces the capacity of global modeling and structural feature enhancement. In the structure-guided mechanism, residual signals provide stable support for feature alignment. Over-compression of these signals harms the completeness of task representations and limits the model's generalization in high-dimensional structural spaces.

Considering the practical needs of task waiting time prediction, the model achieves optimal performance with a medium level of regularization strength, such as 10^{-5} to 10^{-4}. This confirms that under the context modeling and structure-aware framework, moderate regularization is essential for improving model stability and robustness. This setting maintains sufficient expressive power while constraining the model's overreaction to sparse and abnormal features, thus providing more reliable waiting time estimates for scheduling systems.

4.3.4 Analysis of the Impact of Hidden Layer Dimensions on Queue Time Prediction Accuracy

This paper also gives the impact of the hidden layer dimension on the queue time prediction accuracy. The experimental results are shown in Figure 5.

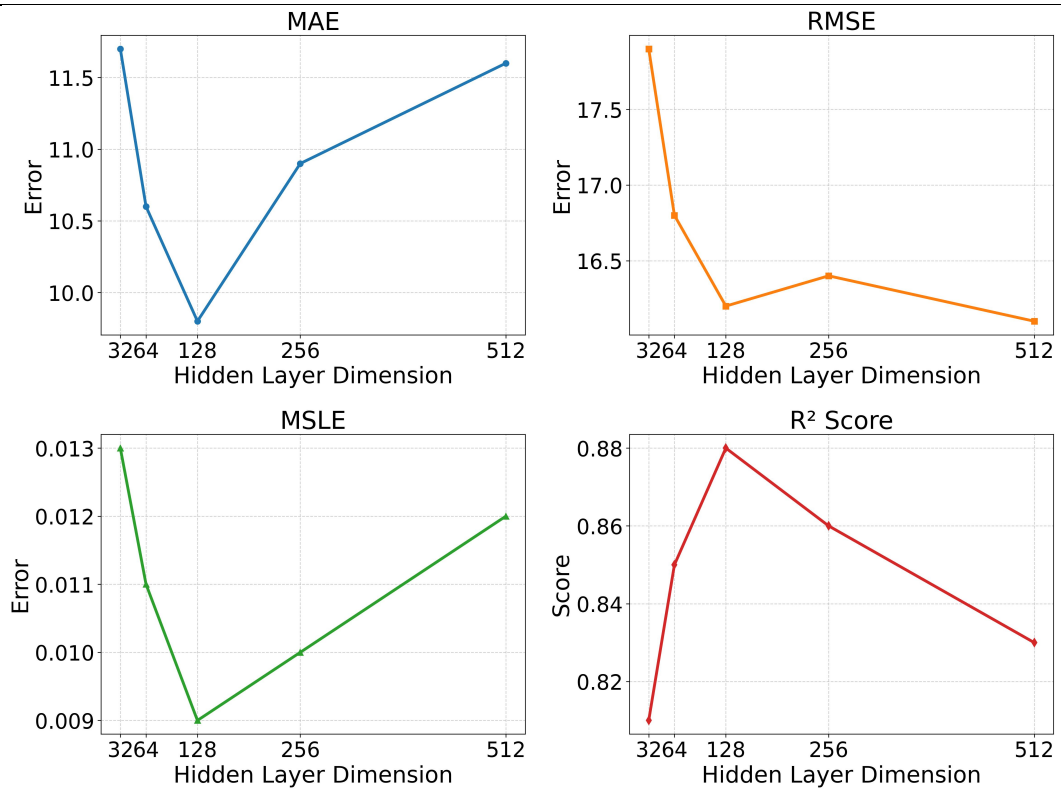

Figure 5. Analysis of the impact of hidden layer dimensions on queue time prediction accuracy

The choice of hidden layer dimensionality has a significant impact on both the expressive power and generalization performance of the waiting time prediction model. When the dimensionality is small, such as 32 or 64, the model contains fewer parameters and has limited capacity. This results in information loss when handling complex scheduling dependencies and dynamic feature interactions, leading to poor performance across several evaluation metrics.

As the hidden dimension increases, the model gains stronger feature modeling ability. It becomes capable of capturing more complex nonlinear patterns and multi-level dependency structures. This leads to noticeable improvements in MAE and MSLE, along with higher R^2 scores. In particular, when the dimension is set to 128, the model achieves the best overall performance. This suggests that this setting provides a good balance between feature richness and structural stability.

When the dimension increases further to 256 or 512, the model has more capacity but also suffers from increased redundancy and a higher risk of overfitting. As a result, error metrics begin to rise, and R^2 scores decline. This indicates that in waiting time prediction tasks, the model's capacity should match the complexity of the task and the scale of the data. Otherwise, feature learning may drift, and generalization performance may decrease.

Overall, an appropriate hidden layer dimension is essential for the coordinated expression of the structure-guided feature enhancement mechanism (SGFE) and the context modeling structure proposed in this study. A suitable dimensionality effectively supports structure-aware alignment and residual feature integration, thereby enhancing the model's ability to represent waiting time variations under dynamic and competitive scheduling scenarios.

5. Conclusion

This study addresses the problem of waiting time modeling in complex scheduling systems. It proposes a prediction framework that integrates context modeling and structure-aware mechanisms. The framework

effectively handles challenges caused by task interference, system uncertainty, and feature heterogeneity. By designing a Global Context-Aware Regression Module and a Structure-Guided Feature Enhancement mechanism, the model captures the underlying relationships between task behaviors and system dynamics. This enables more accurate and stable waiting time predictions.

At the methodological level, the study emphasizes the joint modeling of structural semantics and temporal context. It overcomes the limitations of traditional approaches that rely only on task history or local features. The proposed modules are designed with strong modularity and extensibility. They can be flexibly adapted to various resource management and task scheduling scenarios. This significantly enhances the model's discriminative power and robustness under heavy load fluctuations and complex scheduling conditions.

In terms of practical value, the proposed framework applies to queue management in cloud computing platforms, edge systems, and high-performance scheduling environments. It also offers reliable time-aware support for real-world applications such as smart manufacturing, autonomous driving, resource planning, and data center service orchestration. The model's advantages in accuracy and generalization provide strong support for delay-sensitive optimization, load balancing strategies, and proactive scheduling mechanisms.

In future work, the model's adaptability under more complex conditions could be further explored. These include heterogeneous task topologies, non-stationary system dynamics, and cross-domain deployment. Moreover, integrating federated learning and privacy-preserving mechanisms could extend the framework to collaborative multi-source modeling. This would lay the foundation for building more generalizable, flexible, and task-aware scheduling prediction systems, promoting deeper integration of intelligent scheduling in real-world engineering and smart applications.

References

[1] C. Vercellino, A. Scionti, G. Varavallo et al., "A Machine Learning Approach for an HPC Use Case: The Jobs Queuing Time Prediction," Future Generation Computer Systems, vol. 143, pp. 215-230, 2023.

[2] N. Brown, G. Gibb, E. Belikov et al., "Predicting Batch Queue Job Wait Times for Informed Scheduling of Urgent HPC Workloads," arXiv preprint arXiv:2204.13543, 2022.

[3] S. Li, "Adaptive Scheduling for Multi-Model Collaborative Distributed Inference under Resource Heterogeneity and Dynamic Workloads," 2024.

[4] Qiu, J, "Learning Collaborative and Robust Scheduling Policies for Microservice Backends under Uncertainty," 2024.

[5] A. Al-Mousa, H. Al-Zubaidi and M. Al-Dweik, "A Machine Learning-Based Approach for Wait-Time Estimation in Healthcare Facilities With Multi-Stage Queues," IET Smart Cities, vol. 6, no. 4, pp. 333-350, 2024.

[6] L. Zhou, L. Zhang and B. K. P. Horn, "Deep Reinforcement Learning-Based Dynamic Scheduling in Smart Manufacturing," Procedia CIRP, vol. 93, pp. 383-388, 2020.

[7] Y. Fan, Z. Lan, T. Childers et al., "Deep Reinforcement Agent for Scheduling in HPC," Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 807-816, 2021.

[8] Z. Xu, Y. Gong, Y. Zhou et al., "Enhancing Kubernetes Automated Scheduling With Deep Learning and Reinforcement Techniques for Large-Scale Cloud Computing Optimization," Proceedings of the Ninth International Symposium on Advances in Electrical, Electronics, and Computer Engineering (ISAEECE 2024), vol. 13291, pp. 1595-1600, 2024.

[9] G. Zhou, W. Tian, R. Buyya et al., "Deep Reinforcement Learning-Based Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions," Artificial Intelligence Review, vol. 57, no. 5, Art. no. 124, 2024.

[10] E. Balkanski, N. Perivier, C. Stein et al., "Energy-Efficient Scheduling With Predictions," Advances in Neural Information Processing Systems, vol. 36, pp. 79012-79023, 2023.

[11] T. S. Farias and J. Maziero, "Feature Alignment as a Generative Process," Frontiers in Artificial Intelligence, vol. 5, Art. no. 1025148, 2023.

[12] J. H. Lee and G. Lee, "Feature Alignment by Uncertainty and Self-Training for Source-Free Unsupervised Domain Adaptation," Neural Networks, vol. 161, pp. 682-692, 2023.

[13] A. Baratin, T. George, C. Laurent et al., "Implicit Regularization via Neural Feature Alignment," Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 2269-2277, 2021.

[14] X. Huang and S. H. Choi, "Learning Feature Alignment and Dual Correlation for Few-Shot Image Classification," CAAI Transactions on Intelligence Technology, vol. 9, no. 2, pp. 303-318, 2024.

[15] C. Zhang, "Adaptive Multi-Tenant Resource Scheduling in Cloud Computing via Reinforcement Learning," 2024.

[16] Y. Pei, T. Huang, W. Van Ipenburg et al., "ResGCN: Attention-Based Deep Residual Modeling for Anomaly Detection on Attributed Networks," Machine Learning, vol. 111, no. 2, pp. 519-541, 2022.

[17] H. Yu, L. T. Yang, X. Fan et al., "A Deep Residual Computation Model for Heterogeneous Data Learning in Smart Internet of Things," Applied Soft Computing, vol. 107, Art. no. 107361, 2021.

[18] K. He, X. Zhang, S. Ren et al., "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[19] M. Xu, C. Song, H. Wu et al., "esDNN: Deep Neural Network Based Multivariate Workload Prediction in Cloud Computing Environments," ACM Transactions on Internet Technology, vol. 22, no. 3, pp. 1-24, 2022.

[20] Y. S. Patel and J. Bedi, "MAG-D: A Multivariate Attention Network Based Approach for Cloud Workload Forecasting," Future Generation Computer Systems, vol. 142, pp. 376-392, 2023.

[21] S. Arbat, V. K. Jayakumar, J. Lee et al., "Wasserstein Adversarial Transformer for Cloud Workload Prediction," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 11, pp. 12433-12439, 2022.

[22] A. S. Romanov, A. V. Papadopoulos and N. Herbst, "Uncertainty-Aware Workload Forecasting for Cloud Environments," Proceedings of the 2023 IEEE International Conference on Cloud Computing (CLOUD), pp. 190-198, 2023.