# GACL: Graph Attention Collaborative Learning for Temporal QoS Prediction

Shengxiang Hu, Guobing Zou, Bofeng Zhang, Shaogang Wu, Shiyi Lin, Yanglan Gan, and Yixin Chen, *Fellow, IEEE*

***Abstract*—Accurate prediction of temporal QoS is crucial for maintaining service reliability and enhancing user satisfaction in dynamic service-oriented environments. However, current methods often neglect high-order latent collaborative relationships and fail to dynamically adjust feature learning for specific user-service invocations, which are critical for precise feature extraction within each time slice. Moreover, the prevalent use of RNNs for modeling temporal feature evolution patterns is constrained by their inherent difficulty in managing long-range dependencies, thereby limiting the detection of long-term QoS trends across multiple time slices. These shortcomings dramatically degrade the performance of temporal QoS prediction. To address the two issues, we propose a novel Graph Attention Collaborative Learning (GACL) framework for temporal QoS prediction. Building on a dynamic user-service invocation graph to comprehensively model historical interactions, it designs a target-prompt graph attention network to extract deep latent features of users and services at each time slice, considering implicit target-neighboring collaborative relationships and historical QoS values. Additionally, a multi-layer Transformer encoder is introduced to uncover temporal feature evolution patterns, enhancing temporal QoS prediction. Extensive experiments on the WS-DREAM dataset demonstrate that GACL significantly outperforms state-of-the-art methods for temporal QoS prediction across multiple evaluation metrics, achieving the improvements of up to 38.80%.**

***Index Terms*—Web service, temporal QoS prediction, dynamic user-service invocation graph, target-prompt graph attention network, user-service temporal feature evolution.**

## I. Introduction

In TODAY's interconnected service-oriented architecture, Quality of Service (QoS) metrics serve as critical indicators for maintaining reliability and enhancing user satisfaction in Web services. QoS encompasses several parameters, including response time, throughput, and availability, which directly influence overall user experience [1]. Faster response times enhance user satisfaction, while higher throughput and availability ensure service continuity and stability. With the global acceleration of digital transformation, enterprise and individual dependence on online services has made high service quality essential [2], [3], [4], making effective QoS prediction paramount for service providers to remain competitive and deliver consistent user experiences.

However, accurately predicting QoS metrics remains challenging due to the dynamic and complex nature of network environments [5], [6]. Rapid changes in network traffic, server load fluctuations, and evolving user behavior patterns significantly impact QoS, causing volatility and unpredictability. Static prediction methods [2], [7], [8], which overlook temporal trends, fail to address these issues. Consequently, enhancing model accuracy requires comprehensive understanding of the temporal dynamics underlying these metrics. Temporal QoS prediction addresses these challenges by incorporating a temporal dimension that treats historical QoS data as a time-dependent sequence, thereby capturing inherent patterns and trends. Advances in machine learning and data mining have enabled significant progress in developing these predictive models.

Current research in temporal QoS prediction falls into four main categories: collaborative filtering (CF) with temporal factors [9], [10], [11], sequence prediction analysis [12], [13], tensor decomposition [4], [14], and deep learning [5], [15]. Temporal CF selects similar neighbors using random walk algorithms and vector comparisons to address data sparsity and improve prediction accuracy. Sequence prediction approaches adopt the ARIMA model [12], [13] to enhance prediction performance of missing temporal QoS. Tensor decomposition converts the traditional two-dimensional user-service matrix into a three-dimensional tensor, employing techniques such as CP decomposition [14], personalized gated recurrent units (PGRU), and generalized tensor factorization (GTF) [16]. Deep learning models leverage recurrent neural networks (RNNs) and their variants like LSTM and GRU to forecast unknown QoS using historical invocation data and multidimensional context. These approaches integrate temporal information to detect dynamic patterns in QoS data, effectively addressing network volatility and enhancing prediction precision.

Received 16 October 2024; revised 30 March 2025; accepted 11 May 2025. Date of publication 15 May 2025; date of current version 7 August 2025. This work was supported by National Natural Science Foundation of China (No. 62272290, 62172088), and Shanghai Natural Science Foundation (No. 21ZR1400400). The associate editor coordinating the review of this article and approving it for publication was N. Kamiyama. *(Corresponding authors: Guobing Zou; Bofeng Zhang.)*

Shengxiang Hu, Guobing Zou, Shaogang Wu, and Shiyi Lin are with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China (e-mail: shengxianghu@shu.edu.cn; gbzou@shu.edu.cn).

Bofeng Zhang is with the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China (e-mail: bfzhang@sspu.edu.cn).

Yanglan Gan is with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China (e-mail: ylgan@dhu.edu.cn).

Yixin Chen is with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: chen@cse.wustl.edu).

Digital Object Identifier 10.1109/TNSM.2025.3570464

Despite these advancements, two critical limitations persist in temporal QoS prediction. First, although recent approaches [17], [18], including our previous work [19], have integrated neural graph learning to incorporate indirect user-service invocation relationships, they employ uniform neighbor aggregation strategies that fail to differentiate the importance of neighbors based on their relevance to the target user-service pair. This one-size-fits-all approach neglects the varying influence of neighboring users or services in real-world scenarios, which depends on factors such as invocation frequency, service requirement similarity, and QoS experience consistency. Consequently, these models cannot generate precise, context-aware feature representations that adapt to the specific nuances of individual QoS invocations that comprise the recorded measurements of service quality when a particular user accesses a specific service at a given time in diverse and dynamic service environments. Second, current models predominantly rely on RNNs to capture QoS evolution across time slices, but RNNs struggle with long-range dependencies [19], [20], limiting their capacity to fully utilize historical data and detect long-term QoS trends. This constraint results in less stable predictive performance, particularly when dealing with services exhibiting complex temporal patterns or when predicting QoS over extended time horizons.

To address these limitations and extend our previous work, Dynamic Graph Neural Collaborative Learning (DGNCL) [19], we propose a novel framework: **G**raph **A**ttention **C**ollaborative **L**earning (**GACL**) for temporal QoS prediction. First, we model historical user-service QoS invocations, which refer to historical interactions where users request and receive quality metrics from Web services, as a temporal service ecosystem and transform it into a dynamic user-service invocation graph spanning multiple consecutive time slices, enabling comprehensive modeling of temporal evolution in user-service interactions. Second, we design a target-prompt graph attention network for fine-grained, invocation-specific feature extraction of each distinct user-service pair. This network dynamically adapts to specific invocation scenarios by employing a novel target-prompt attention strategy that simultaneously leverages indirect invocation relationships and implicit collaborative correlations between targets and their neighbors. By recognizing varying neighbor relevance and adaptively adjusting attention weights during aggregation, our model precisely captures the unique characteristics of each user-service pair (e.g., user preferences, service performance patterns, and their interaction history) for every prediction task. Finally, to capture long-term temporal dependencies, we implement a multi-layer Transformer [21] encoder that effectively models the temporal evolution of dynamically learned user and service features across extended time horizons, resulting in highly accurate temporal QoS forecasts.

The main contributions of this paper are:

- We propose GACL for temporal QoS prediction with two key innovations: target-prompt graph attention for context-aware feature extraction and a Transformer encoder for modeling long-term dependencies, together overcoming data sparsity challenges in complex temporal service ecosystems.
- We introduce a target-prompt attention mechanism that adapts to each specific invocation by weighing neighbor importance based on collaborative relationships and historical QoS patterns, enhancing feature extraction in sparse service environments.
- Experiments on the WS-DREAM dataset demonstrate GACL outperforms ten baseline methods with up to 38.80% accuracy improvement across multiple metrics.

## II. Problem Definition

*Definition 1 (Temporal Service Ecosystem):* A temporal service ecosystem can be defined as a four-tuple $\xi = \langle \mathcal{U}, \mathcal{S}, \mathcal{T}, \mathcal{I} \rangle$, where $\mathcal{U} = \{u_i\}_{i=1}^{n}$ represents a set of $n$ users, $\mathcal{S} = \{s_j\}_{j=1}^{m}$ denotes a set of $m$ Web services, and $\mathcal{T} = \{t_1, t_2, \dots\}$ is a set of continuous time slices. The set $\mathcal{I} = \{\tau_{u,s}^{t}\}$ comprises user-service QoS invocations across multiple temporal slices.

The temporal service ecosystem captures the dynamic interactions between users and services, presenting them within a temporal context to reflect interaction patterns and QoS performance across different time slices. A specific user-service QoS invocation in the ecosystem is defined as follows:

*Definition 2 (User-Service QoS Invocation):* Within a temporal service ecosystem $\xi$, a user-service QoS invocation is represented by a four-tuple $\tau = \langle u, s, t, r_{us}^{t} \rangle$, where $u \in \mathcal{U}$ denotes a user, $s \in \mathcal{S}$ is a Web service, $t \in \mathcal{T}$ represents a time slice, and $r_{us}^{t}$ is the QoS value when $u$ invokes $s$ at $t$.

Given a sequence of consecutive time slices and their corresponding user-service invocations, the problem of temporal QoS prediction can be formally defined as follows:

*Definition 3 (Temporal QoS Prediction):* Given a temporal service ecosystem $\xi$ and the associated historical QoS matrix sequence $\mathcal{R} = \{R^t \in \mathbb{R}^{n \times m}\}_{t=1}^{|\mathcal{T}|}$, where $R^t$ is the historical QoS matrix at time slice $t$ and $r_{us}^{t}$ denotes the QoS value for a user $u$ and a service $s$ in $R^t$, the temporal QoS prediction problem aims to learn interaction patterns between users and services and their temporal evolution to accurately predict the QoS for user-service invocations at subsequent time slices. It can be formally defined as:

$$\hat{R}^{t+1} = f\left(R^1, R^2, \dots, R^{|\mathcal{T}|} | \Theta_f\right) \tag{1}$$

where $\hat{R}^{t+1} \in \Re^{n \times m}$ is the predicted QoS matrix for time slice $t+1$, based on the historical QoS data $\mathcal{R}$. The function $f(\cdot|\Theta_f)$ represents the proposed prediction framework, with $\Theta_f$ being the learned parameters.

This indicates that when a target user $u \in \mathcal{U}$ invokes a target service $s \in \mathcal{S}$ at time slice $t+1$, the predicted QoS value is $\hat{r}_{us}^{t+1} \in \hat{R}^{t+1}$.

## III. Approach

Figure 1 illustrates the overall architecture of the GACL framework, which operates in four key phases: First, it transforms a temporal service ecosystem $\xi$ with $|\mathcal{T}|$ time slices into a discrete dynamic user-service invocation graph, where each snapshot represents user-service invocations and their corresponding QoS values for a specific time slice $t$.

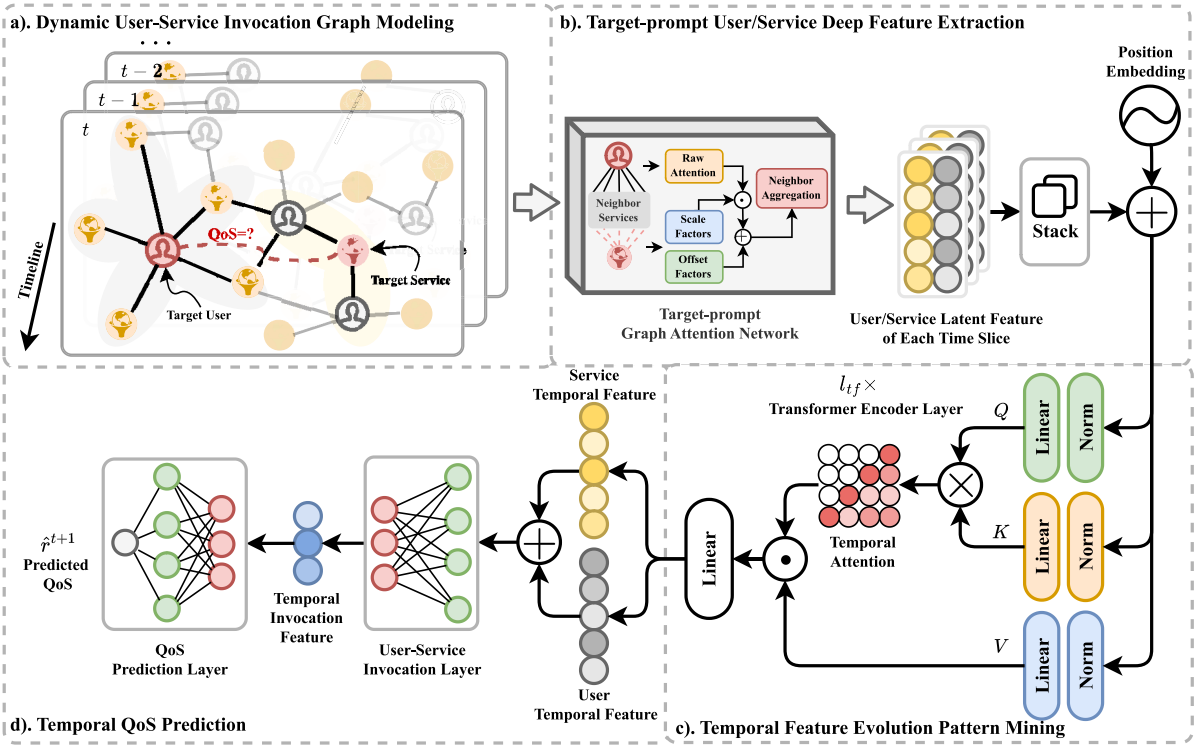



Fig. 1. Overview of the GACL framework for temporal QoS prediction.

Second, for a target invocation $\tau$, a target-prompt graph attention network processes the invocation graph of each time slice, extracting user and service features through biased neighbor aggregation that simultaneously considers implicit collaborative relationships and historical QoS values. Third, user and service features from different time slices are fed into a multi-layer Transformer encoder to capture temporal evolution dynamics, generating temporal features for time slice $t+1$. Finally, two multi-layer perceptrons (MLPs) extract latent temporal user-service invocation features and predict QoS values.

### A. Dynamic User-Service Invocation Graph Modeling

Discrete dynamic graphs can effectively model complex interactions and temporal variations between entities [22], [23]. Therefore, we first model the temporal service ecosystem $\xi$ with $|\mathcal{T}|$ time slices as a dynamic user-service invocation graph. Each snapshot of this dynamic graph represents user-service invocations and the corresponding QoS records for a specific time slice. In this dynamic graph, users and services are represented as nodes. Any historical user-service invocation is recorded as an edge between the target user and service in the snapshot of the appropriate time slice, with the QoS value as the edge weight. The formal definition of this dynamic invocation graph is as follows:

*Definition 4 (Dynamic User-Service Invocation Graph):* A dynamic user-service invocation graph is formulated as $\mathcal{G} = \{\mathcal{G}^t\}_{t=1}^{|\mathcal{T}|}$. For each snapshot $\mathcal{G}^t = \langle \mathcal{V}_u, \mathcal{V}_s, \mathcal{E}^t, \mathcal{W}^t \rangle$, it is a bipartite graph transformed from a sub temporal service ecosystem $\xi^t = \langle \mathcal{U}, \mathcal{S}, t, \mathcal{I}^t \rangle$ and the corresponding QoS matrix $R^t$ at time slice $t$. Here, $\mathcal{V}_u = \{v_{u_i}\}_{i=1}^{n}$ is a set of $n$ user vertices; $\mathcal{V}_s = \{v_{s_j}\}_{j=1}^{m}$ is a set of $m$ service vertices; $\mathcal{E}^t$ is a set of edges representing user-service invocation relationships. If $r_{u_i s_j}^t \in R^t$, there exists an edge $e_{ij}^t = e_{ji}^t \in \mathcal{E}^t$ between $v_{u_i} \in \mathcal{V}_u$ and $v_{s_j} \in \mathcal{V}_s$; $\mathcal{W}^t$ is a set of edge weights. If $e_{ij}^t \in \mathcal{E}^t$, there exists a corresponding weight $w_{ij}^t \in \mathcal{W}^t$, which can be derived from $r_{u_i s_j}^t \in R^t$.

To initialize the features of each node in $\mathcal{G}$, we apply a random embedding approach. Specifically, each node $v \in \mathcal{V}_u \cup \mathcal{V}_s$ is assigned a unique ID, which is then embedded into a $d_e$-dimensional latent space to generate its initial feature, denoted as $\mathbf{e}_v$. This embedding process can be formally expressed as:

$$\mathbf{e}_v = Embedding(v) \tag{2}$$

where $Embedding(\cdot)$ represents the embedding function that maps the unique ID of node $v$ to a $d_e$-dimensional vector. These embeddings serve as the initial features for the nodes in the dynamic user-service invocation graph, capturing the inherent characteristics of users and services based on their unique identifiers.

Through these steps, we construct the dynamic user-service invocation graph $\mathcal{G}$, which will be used for subsequent target-prompt deep feature learning for users and services.

### B. Target-Prompt Deep Latent Feature Extraction of Users and Services

Given the constructed dynamic user-service invocation graph $\mathcal{G}$ and a target invocation $\langle u, s, t+1 \rangle$ whose QoS requires prediction, we introduce a novel target-prompt graph

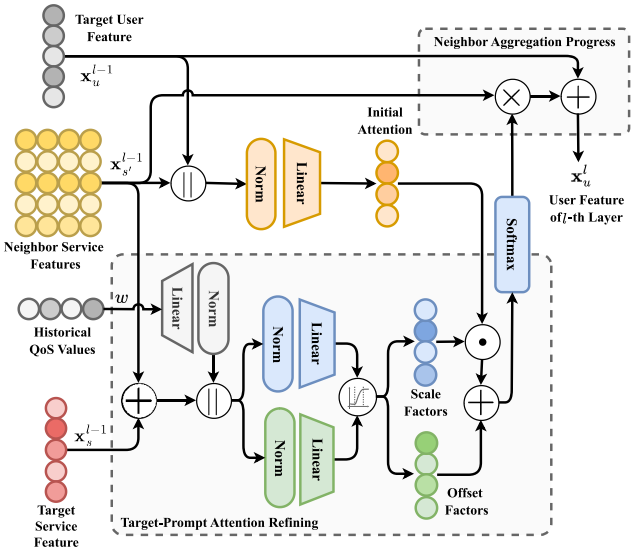



Fig. 2. Message passing and aggregation in the target-prompt graph attention network.

attention network to extract the deep latent features of the target user $u$ and service $s$ at each previous time slice.

Building on our prior work [19], we observe that a user's features are influenced by both direct service invocations and indirect interactions with non-adjacent entities. Similarly, a service's features emerge from both direct user interactions and collaborative relationships with other services. Such complex relationships can be effectively captured through neighbor information aggregation using the multi-layer recursive message passing mechanism of Graph Neural Networks (GNNs) [24] within the invocation graph $\mathcal{G}^t$ for each time slice $t$. We first detail the feature extraction process for target users as depicted in Figure 2, noting that an identical methodology applies to target services.

In our approach, neighbors contribute differentially to feature extraction based on contextual relevancy, as users and services with similar contexts typically indicate comparable physical network environments. We therefore adopt the graph attention strategy from GAT [25], computing the initial semantic aggregation attention between target nodes and their neighbors based on contextual semantic relevancy. Higher attention values indicate greater influence on feature extraction. Specifically, for a target user $u$, $\mathcal{N}_u^t \subseteq \mathcal{V}_s$ denotes the adjacent service vertices directly connected to $u$ in $\mathcal{G}^t$ (i.e., $u$'s first-hop service neighbors at time slice $t$). For each neighbor service $s' \in \mathcal{N}_u^t$, the initial semantic aggregation attention is calculated as:

$$\left(attn_{u\leftarrow s'}^t\right)^l = \sigma_{sigmoid}\left(W_{attn}^l\left(\mathbf{x}_u^{l-1} + \mathbf{x}_{s'}^{l-1}\right)^T\right) \quad (3)$$

where $\sigma_{sigmoid}(\cdot)$ denotes the sigmoid activation function, $W_{attn}^l \in \mathbb{R}^d$ represents learnable attention parameters for the $l^{th}$ message passing layer, $\mathbf{x}_u^{l-1} \in \mathbb{R}^d$ is the hidden vector of the target user output from the $(l-1)^{th}$ layer, and $d$ is the dimensionality of the latent features. The term $(attn_{u\leftarrow s'}^t)^l \in \mathbb{R}$ represents the initial semantic attention value calculated at the $l^{th}$ layer. When $l = 1$, this initial semantic attention is derived from the embeddings of users and services:

$$\mathbf{x}_v^{l-1} = \mathbf{e}_v, \quad v \in \left\{u \cup \mathcal{N}_u^t\right\} \quad \text{and} \quad l = 1 \quad (4)$$

Drawing from collaborative filtering principles [2], [3], we recognize that neighboring services closely related to a target service contribute significantly to accurate QoS prediction. Consequently, extracting features for the target user requires prioritizing these similar neighboring services. Furthermore, historical QoS values, represented as edge weights in the invocation graph, directly reflect the actual network state and physical environment context between users and services. These values are crucial for accurately capturing interaction dynamics and must be incorporated during feature aggregation. Existing methods rarely integrate these critical factors simultaneously, resulting in suboptimal feature learning and reduced prediction accuracy.

To address these limitations, we propose a novel target-prompt attention strategy (illustrated in the bottom right of Figure 2) that enhances target user feature extraction by simultaneously accounting for implicit collaborative relevance between neighboring and target services and corresponding historical QoS records. We model this enhancement as an affine transformation that adjusts the initial semantic attention based on a prompt of the target service and historical QoS data:

$$\hat{\mathbf{x}}_{s'}^{l-1} = \left[norm\left(\mathbf{x}_s^{l-1} + \mathbf{x}_{s'}^{l-1}\right) || norm\left(W_w^l w_{us'}^t + b_w^l\right)\right] \quad (5)$$

$$\left(\alpha_{u\leftarrow s'}^t\right)^l = \sigma_{tanh}\left(W_\alpha^l\left(\hat{\mathbf{x}}_{s'}^{l-1}\right)^T + b_\alpha^l\right) \quad (6)$$

$$\left(\beta_{u\leftarrow s'}^t\right)^l = \sigma_{tanh}\left(W_\beta^l\left(\hat{\mathbf{x}}_{s'}^{l-1}\right)^T + b_\beta^l\right) \quad (7)$$

$$\left(\hat{attn}_{u\leftarrow s'}^t\right)^l = \left(\alpha_{u\leftarrow s'}^t\right)^l * \left(attn_{u\leftarrow s'}^t\right)^l + \left(\beta_{u\leftarrow s'}^t\right)^l \quad (8)$$

where $(\alpha_{u\leftarrow s'}^t)^l, (\beta_{u\leftarrow s'}^t)^l \in (-1, 1)$ represent learned scaling and shifting factors for the initial semantic attention. $\sigma_{tanh}(\cdot)$ is the hyperbolic tangent activation function, $||$ denotes vector concatenation, $W_w^l \in \mathbb{R}^d$, $W_\alpha^l, W_\beta^l \in \mathbb{R}^{2d}$, and $b_w^l, b_\alpha^l, b_\beta^l$ are learnable parameters. The resulting adjusted attention value $(\hat{attn}_{u\leftarrow s'}^t)^l$ captures three key aspects: contextual semantic relevance between the target user and neighboring services, implicit collaborative relevance between the target service and neighboring services, and the influence of historical QoS values on neighbor feature contributions.

We then employ biased message propagation and aggregation based on these target-prompt attention values to enhance feature extraction for the target user, as shown in the top right corner of Figure 2. Specifically, the message $(\mathbf{m}_{u\leftarrow s'}^t)^l \in \mathbb{R}^d$ transferred from a neighboring service $s' \in \mathcal{N}_u^t$ to user $u$ is formulated as:

$$\left(\mathbf{m}_{u\leftarrow s'}^t\right)^l = \frac{exp\left(\left(\hat{attn}_{u\leftarrow s'}^t\right)^l\right)}{\sum_{i\in\mathcal{N}_u^t} exp\left(\left(\hat{attn}_{u\leftarrow i}^t\right)^l\right)} W_{msg}^l \mathbf{x}_{s'}^{l-1} \quad (9)$$

Here, $W_{msg}^{l} \in \mathbb{R}^{d\times d}$ represents a trainable weight matrix for the $l^{th}$ layer. Messages from all adjacent services are aggregated as:

$$\left(\mathbf{x}_u^t\right)^l = \sigma_{prelu}\left((\mathbf{x}_u^t)^{l-1} + \sum_{s'\in\mathcal{N}_u^t}(\mathbf{m}_{u\leftarrow s'}^t)^l\right) \tag{10}$$

where $(\mathbf{x}_u^t)^l \in \mathbb{R}^d$ represents the aggregated features of $u$, incorporating first-order messages that capture behavioral features from directly invoked services. The function $\sigma_{prelu}$ denotes the PReLU [26] activation function.

By stacking $l_g$ message-passing layers, we extend the aggregation to encompass messages from $l_g$-hop user and service neighbors, culminating in sophisticated invocation-specific high-order latent features $(\mathbf{x}_u^t)^{l_g}$ for user $u$. This advanced representation captures latent invocation correlations between $u$ and non-invoked services, as well as collaborative relationships among structurally proximal user neighbors. The process for extracting high-order latent features $(\mathbf{x}_s^t)^{l_g}$ for a target service $s$ mirrors that of user $u$, differing only in the neighbor user message propagation and aggregation phase, where the target prompt module modifies the initial semantic attention based on implicit collaborative relevance between neighboring users and the target user.

With a window size $ws \in \mathbb{N}^+$, we extract deep latent features for both target user and service from user-service invocation graphs over the $ws$ preceding time slices leading to the target time slice $t+1$. This yields sequence features $\{(\mathbf{x}_u^i)^{l_g}\}_{i=t-ws+1}^{t}$ for users and $\{(\mathbf{x}_s^i)^{l_g}\}_{i=t-ws+1}^{t}$ for services. These sequences enable analysis of complex nonlinear evolutionary patterns of the target user and service over time, providing insights into their dynamic interactions and potential future behaviors.

In this stage, our target-prompt mechanism integrates historical QoS values directly as edge weights in the graph, allowing real network states to explicitly influence feature learning within each time slice. This design ensures that extracted features accurately reflect the actual service performance conditions of their specific temporal context.

### C. Temporal Feature Evolution Mining of Users and Services

Network state fluctuations and invocation behavior changes manifest as complex nonlinear evolution patterns in user and service features over time. To enhance temporal QoS prediction accuracy, we harness the Transformer architecture's [21] capability to process long sequences, extract these intricate evolution patterns, and generate temporal features that encapsulate these patterns.

We first consolidate the user and service feature sequences into respective feature matrices:

$$X_u = stack\left(\left\{\left(\mathbf{x}_u^i\right)^{l_g}\right\}_{i=t-ws+1}^{t}\right) \tag{11}$$

$$X_s = stack\left(\left\{\left(\mathbf{x}_s^i\right)^{l_g}\right\}_{i=t-ws+1}^{t}\right) \tag{12}$$

The self-attention mechanism alone cannot effectively distinguish temporal positions within these matrices, potentially compromising the chronological integrity of features across time slices. Therefore, following standard Transformer design principles, we incorporate positional encoding to explicitly embed temporal information. For notational clarity, we abstract both target user and service feature matrices as $X$, with the encoding process defined as:

$$PE_{pos,2i} = sin\left(\frac{pos}{10000^{2i/d}}\right) \tag{13}$$

$$PE_{pos,2i+1} = cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{14}$$

$$\hat{X} = X + PE \tag{15}$$

where $PE \in \mathbb{R}^{ws\times d}$ represents the positional encoding matrix, $pos$ denotes the row index in the feature matrix, and $i$ indicates the column index within each row vector.

We then process $\hat{X}$ through a Transformer encoder with $l_{tf}$ layers and $l_{hd}$ attention heads to extract temporal evolution patterns and generate features for time slice $t+1$. In each encoder layer, multi-head attention captures temporal dependencies across features from different time slices. For the $i$-th layer, the computation proceeds as:

$$Z^0 = \hat{X} \tag{16}$$

$$\begin{aligned} Z'^i &= MultiHead\left(Z^{i-1}\right) \\ &= Concat\left(head_1^i, head_2^i, \ldots, head_{l_{hd}}^i\right) W_{hd}^i \end{aligned} \tag{17}$$

Each attention head $head_j^i$ is computed as:

$$head_j^i = softmax\left(\frac{Q_j^i\left(K_j^i\right)^T}{\sqrt{d_k}} V_j^i\right) \tag{18}$$

$$Q_j^i = Z^{i-1}\left(W_j^Q\right)^i \tag{19}$$

$$K_j^i = Z^{i-1}\left(W_j^K\right)^i \tag{20}$$

$$V_j^i = Z^{i-1}\left(W_j^V\right)^i \tag{21}$$

where $W_j^Q$, $W_j^K$, and $W_j^V$ are learnable weight matrices in the self-attention mechanism. $Q_j^i$, $K_j^i$ and $V_j^i$ represent the query, key, and value matrices, respectively. Higher attention values in specific dimensions indicate greater contributions from corresponding time slices toward generating $t+1$ temporal features. The scaling factor $\sqrt{d_k}$ normalizes dot product results as prescribed in [21].

Following the multi-head attention, a feed-forward network (FFN) [21] performs nonlinear transformation:

$$\hat{Z}^i = FFN\left(Z'^i\right) = ReLU\left(Z'^i W_1^i + b_1^i\right) W_2^i + b_2^i \tag{22}$$

$$Z^i = LayerNorm\left(\hat{Z}^i + Z^{i-1}\right) \tag{23}$$

where $W_1^i$, $W_2^i$, $b_1^i$, and $b_2^i$ are trainable FFN parameters. After processing through $l_{tf}$ Transformer encoder layers, we extract the temporal features for $t+1$ as:

$$\mathbf{h}_u^{t+1} = Z_u^{l_{tf}}[-1] \tag{24}$$

$$\mathbf{h}_s^{t+1} = Z_s^{l_{tf}}[-1] \tag{25}$$

We select the final row of the hidden feature matrix $Z^{l_{tf}}$, which encapsulates information from all *ws* historical time slices, as the temporal features of the target user and service, $\mathbf{h}_u^{t+1} \in \mathbb{R}^d$ and $\mathbf{h}_s^{t+1} \in \mathbb{R}^d$, for subsequent QoS prediction.

Unlike graph attention networks that operate within individual time slices, our Transformer approach captures temporal evolution across the entire window. Here, historical QoS information is implicitly embedded in the user and service features themselves, rather than explicitly as edge weights. This complementary design enables our model to effectively learn both spatial relationships and temporal dependencies of QoS patterns.

### D. Temporal QoS Prediction and Model Training

To effectively model the complex non-linear interaction between users and services while maintaining model interpretability, we employ a two-stage MLP design. This hierarchical approach first learns an intermediate invocation feature $\mathbf{h}_{inv}^{t+1}$ that captures the joint interaction patterns, before using a second MLP to map this rich representation to QoS values, enabling more precise feature fusion and improving prediction accuracy. Based on the target-prompt temporal features of $\mathbf{h}_u^{t+1}$ and $\mathbf{h}_s^{t+1}$, we concatenate them and feed the result into an MLP-based neural invocation layer to obtain the invocation feature of $u$ and $s$ at $t+1$:

$$\mathbf{h}_{us}^{t+1} = \mathbf{h}_u^{t+1} || \mathbf{h}_s^{t+1} \tag{26}$$

$$\mathbf{h}_{inv}^{t+1} = \sigma_{prelu}\Big( W_{inv}\mathbf{h}_{us}^{t+1} + b_{inv} \Big) \tag{27}$$

where $||$ denotes the concatenation operation, $W_{inv} \in \mathbb{R}^{d\times 2d}$ and $b_{inv} \in \mathbb{R}$ are the trainable parameters of the MLP.

Consequently, based on the evolutionary invocation feature $\mathbf{h}_{inv}^{t+1} \in \mathbb{R}^d$, we predict the missing QoS $\hat{r}_{us}^{t+1}$ at $t+1$ using a fully connected neural network. The output layer is calculated as follows:

$$\hat{r}_{us}^{t+1} = \sigma_{relu}\Big( W_o\mathbf{h}_{inv}^{t+1} + b_o \Big) \tag{28}$$

where $W_o \in \mathbb{R}^d$ and $b_o \in \mathbb{R}$ are the trainable output parameters, and $\hat{r}_{us}^{t+1}$ is the predicted QoS of the target invocation $\langle u, s, t+1\rangle$.

To train and optimize the model parameters, we use the Mean Square Error (MSE) as the loss function, defined as:

$$\mathcal{L} = \frac{\sum_{u\in U}\sum_{s\in S}\left(\hat{r}_{us}^{t+1} - r_{us}^{t+1}\right)^2}{|U| \times |S|} + \lambda|\Theta|_2^2 \tag{29}$$

where $U$ and $S$ represent the user and service sets, respectively. $\Theta$ denotes all the trainable parameters of our proposed model, and $\lambda$ controls the $L2$ regularization strength to prevent overfitting. We then use mini-batch AdamW [27] to update and optimize the parameters.

### E. Time Complexity Analysis

We analyze the computational complexity of GACL for predicting QoS of a specific user-service pair $\langle u, s, t+1\rangle$ through its main processing stages. In the feature learning stage, the target-prompt graph attention network processes $k$ neighbors on average for each node across $l_g$ layers. The dominant operations include attention computation ($O(k \cdot d)$) and message transformation with matrix-vector multiplications ($O(k \cdot d^2)$), yielding a theoretical complexity of $O(ws \cdot l_g \cdot k \cdot d^2)$ across *ws* time slices. Through parallel processing of independent time slices, this can be reduced to $O(l_g \cdot k \cdot d^2)$ in practice.

For temporal pattern mining, the Transformer encoder with $l_{tf}$ layers requires $O(l_{tf} \cdot ws^2 \cdot d)$ operations due to the self-attention mechanism's quadratic dependency on sequence length. The final QoS prediction through MLPs adds $O(d^2)$ complexity. In real-world deployments where $k$ is small (sparse invocation networks), $d$ is moderate (feature dimension), and *ws* is limited (temporal window size), GACL achieves efficient inference with $O(l_g \cdot k \cdot d^2 + l_{tf} \cdot ws^2 \cdot d)$ complexity, enabling real-time QoS prediction while maintaining high accuracy.

TABLE I
THE STATISTICS OF WS-DREAM DATASET

| Item | Value | |
|---|---|---|
| Name of Sub-Dataset | Response Time (RT) | Throughput (TP) |
| # of Users | 142 | |
| # of Services | 4,500 | |
| # of Service Invocations | 27,392,643 | |
| Range of QoS | 0-20s | 0-6727kbps |
| Mean & Variance of QoS | 3.2±6.1 | 11.3±54.3 |
| # of Time Slices | 64 | |
| Overall Sparsity | 66.98% | |

## IV. EXPERIMENTS

### A. Experimental Setup and Dataset

Our experimental framework was validated on a workstation equipped with an NVIDIA GTX 4090 GPU, an Intel Xeon Gold 6130 CPU, and 1 TB of RAM. The experimental suite for GACL was developed using Python 3.9.6, PyTorch 2.0.1, and CUDA 12.0 to ensure compatibility and optimized performance.

*1) Dataset:* To evaluate the effectiveness of GACL, we conducted extensive experiments on the publicly available WS-DREAM dataset [1], which is the most widely-adopted, largest-scale real-world benchmark in the temporal QoS prediction domain that has become the de facto standard for research [4], [19], [20], [28], [29], including two types of QoS criteria: response time (RT) and throughput (TP). It comprises 142 independent users, 4500 Web services, and a total of 27,392,643 user-service invocations for each QoS criterion, partitioned into 64 temporal groups of historical QoS records.

The overall QoS sparsity, defined as the percentage of observed historical QoS values in all possible user-service-timeslice combinations, of the WS-DREAM dataset is approximately 66.98%. To rigorously evaluate our model's robustness under different sparsity conditions reflective of real-world scenarios, we conducted experiments at four distinct density levels: 5%, 10%, 15%, and 20%. Table I summarizes the detailed statistics.

*2) Experimental Setup:* We designed a sampling strategy that involved a temporal window of $ws+1$ consecutive time slices, where the first *ws* slices were used as historical

TABLE II
VARIOUS PARAMETER SETTINGS IN THE EXPERIMENTS

| Parameter | Description | Values |
|---|---|---|
| $l_g$ | layers of target-prompt graph attention network | [1,2,3] |
| $d$ | dimension of the temporal features of users and services | [32,64,128,256,512] |
| $\rho$ | density of training dataset $\mathcal{D}_{tr}$ | [5%,10%,15%,20%] |
| $l_{tf}$ | layers of transformer encoder | [1,2,4,8] |
| $l_{hd}$ | head number of multi-head attention mechenism | [1,2,4,8] |
| $ws$ | window size of historical QoS records | [1,4,8,16,32] |
| $lr$ | initial learning rate of AdamW optimizer | 0.001 |
| $bs$ | batch size for training | 64 |
| $ep$ | maximum number of epochs | 100 |
| $es$ | early stopping patience | 5 |
| $wd$ | weight decay coefficient for L2 regularization | 0.0001 |
| $mlp_1$ | structure of first MLP with ReLU activation | [$2d$, $d$] |
| $mlp_2$ | structure of second MLP for QoS prediction | [$d$, 1] |

observations and the last slice as the prediction target. For each of these $ws + 1$ time slices, we randomly sampled subsets of user-service invocations at various densities. For the first $ws$ historical time slices, these sampled invocations provided the observable historical QoS records used as input features. For the last time slice (prediction target), the sampled invocations were used as training labels in $\mathcal{D}_{tr}$, while the remaining unsampled invocations within this same time slice were allocated to the test set $\mathcal{D}_{test}$ as test labels.

Moreover, within the GACL framework, we meticulously tuned various combinations of key parameters, which are detailed in Table II. For model training, we employed the AdamW optimizer with an initial learning rate of 0.001 and a cosine annealing learning rate scheduler. The training process used a batch size of 64 and continued for a maximum of 100 epochs with an early stopping mechanism that terminated training if the validation loss did not improve for 5 consecutive epochs. To prevent overfitting, we applied L2 regularization with a weight decay coefficient of 0.0001. In the QoS prediction component, the first MLP consisted of a two-layer network with dimensions [$2d$, $d$] using ReLU activation, while the second MLP employed a simpler two-layer structure [$d$, 1] to map the invocation features directly to QoS values. The optimal parameter combination was selected for final experiments.

### B. Competing Methods and Evaluation Metrics

*1) Competing Methods:* To evaluate the performance of our proposed GACL for temporal QoS prediction, we compare it with nine competing methods and our previous proposed one DGNCL. They are described as below:

- UPCC [2]: A user-based method using Pearson Correlation Coefficient to calculate user neighborhoods and predict QoS through deviation migration.
- IPCC [3]: A service-based method that predicts QoS using service neighborhoods and combines average QoS with deviation migration.
- WSPred [4]: It extends 2D matrix factorization to 3D tensor representation for temporal QoS prediction.
- PNCF [29]: It utilizes neural networks to learn non-linear user-service relationships from sparse vectors.
- PLMF [15]: It encodes user-service-time relationships using one-hot encoding and LSTM for temporal QoS prediction.
- RTF [16]: It combines Personalized GRU and Generalized Tensor Factorization to analyze long-term and short-term dependency patterns.
- TUIPCC [11]: It integrates temporal values with CF-based QoS values using a temporal similarity computation mechanism.
- RNCF [28]: It incorporates multi-layer GRU into neural collaborative filtering to learn temporal features.
- GAFC [20]: It uses probabilistic matrix factorization, gated feature extraction network, enhanced GRU, and GAN for temporal QoS prediction.
- DGNCL [19]: It introduces a dynamic graph neural collaborative learning framework that integrates user-service invocation graph modeling with graph convolutional networks to capture high-order latent features, while utilizing multi-layer GRU to extract temporal evolution patterns for QoS prediction.

*2) Evaluation Metrics:* For our temporal QoS prediction task, which is fundamentally a regression problem, we employ three complementary evaluation metrics: Mean Absolute Error (MAE), Normalized Mean Absolute Error (NMAE), and Root Mean Squared Error (RMSE). These metrics, defined by Equations (30)-(32), collectively quantify prediction accuracy by measuring deviations between predicted and actual QoS values, with smaller values indicating superior performance.

$$MAE = \frac{\sum_{(u,s)\ \in\mathcal{D}} |\hat{r}_{us}^{t+1} - r_{us}^{t+1}|}{|\mathcal{D}|} \tag{30}$$

$$NMAE = \frac{MAE \cdot |\mathcal{D}|}{\sum_{(u,s)\ \in\mathcal{D}} r_{us}^{t+1}} \tag{31}$$

$$RMSE = \sqrt{\frac{\sum_{(u,s)\ \in\mathcal{D}} \left(\hat{r}_{us}^{t+1} - r_{us}^{t+1}\right)^2}{|\mathcal{D}|}} \tag{32}$$

where $\hat{r}_{us}^{t+1}$ and $r_{us}^{t+1}$ represent predicted and actual QoS values respectively when user $u$ invokes service $s$ at time $t + 1$, and $\mathcal{D}$ denotes the prediction sample set. While MAE provides straightforward error magnitude assessment, NMAE offers cross-dataset comparability through normalization, and RMSE, with its squared term, emphasizes larger errors and outliers—critical for evaluating robustness across diverse prediction scenarios.

### C. Competing Results and Analyses

Table III presents the comparative experimental results of our proposed GACL framework and various baseline methods on the RT and TP datasets under different data density settings (5%, 10%, 15%, 20%). The best results are highlighted in bold, while the second-best results are shaded in gray. As shown in the table, our proposed GACL consistently achieves

TABLE III
COMPARISON RESULTS OF VARIOUS COMPETING MODELS ON RT AND TP DATASETS UNDER DIFFERENT DATA DENSITY SETTINGS

| Dataset | Methods | Density=5% | | | Density=10% | | | Density=15% | | | Density=20% | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **MAE** | **NMAE** | **RMSE** | **MAE** | **NMAE** | **RMSE** | **MAE** | **NMAE** | **RMSE** | **MAE** | **NMAE** | **RMSE** |
| **RT** | UPCC | 0.9022 | 0.2819 | 1.9243 | 0.9587 | 0.2996 | 1.7961 | 0.8948 | 0.2796 | 1.7041 | 0.8513 | 0.2660 | 1.6284 |
| | IPCC | 1.0657 | 0.3330 | 2.0001 | 0.8938 | 0.2793 | 1.7465 | 0.8432 | 0.2635 | 1.6807 | 0.8075 | 0.2523 | 1.6228 |
| | PNCF | 1.1653 | 0.3642 | 1.8358 | 1.0891 | 0.3403 | 1.7221 | 1.0427 | 0.3258 | 1.6533 | 1.0129 | 0.3165 | 1.6170 |
| | WSPred | 0.7809 | 0.2440 | 1.7065 | 0.6894 | 0.2154 | 1.6334 | 0.6726 | 0.2102 | 1.6076 | 0.6634 | 0.2073 | 1.5930 |
| | PLMF | 0.7267 | 0.2271 | 1.7059 | 0.6786 | 0.2121 | 1.6126 | 0.6582 | 0.2057 | 1.5749 | 0.6444 | 0.2014 | 1.5523 |
| | RTF | 0.6681 | 0.2088 | 1.7323 | 0.6302 | 0.1969 | 1.6661 | 0.6120 | 0.1912 | 1.6343 | 0.5352 | 0.1672 | 1.6361 |
| | TUIPCC | 0.7314 | 0.2285 | 1.7760 | 0.6237 | 0.1949 | 1.6323 | 0.8193 | 0.2560 | 2.0587 | 0.6966 | 0.2176 | 1.6351 |
| | RNCF | 0.6920 | 0.2162 | 1.7582 | 0.6007 | 0.1877 | 1.6685 | 0.5902 | 0.1844 | 1.6035 | 0.5559 | 0.1737 | 1.5935 |
| | GAFC | 0.6318 | 0.1974 | 1.5555 | 0.6027 | 0.1883 | 1.6347 | 0.5511 | 0.1722 | 1.4230 | 0.4936 | 0.1542 | 1.5874 |
| | DGNCL | 0.5743 | 0.1795 | 1.2841 | 0.5260 | 0.1643 | 1.1934 | 0.4891 | 0.1528 | 1.1580 | 0.4618 | 0.1443 | 1.1233 |
| | GACL | **0.5126** | **0.1602** | **1.1291** | **0.4781** | **0.1328** | **1.0321** | **0.4439** | **0.1233** | **0.9907** | **0.4216** | **0.1171** | **0.9500** |
| | Gains | 18.87% | 18.84% | 27.41% | 20.41% | 29.25% | 36.00% | 19.45% | 28.40% | 30.38% | 14.59% | 24.06% | 38.80% |
| **TP** | UPCC | 4.0099 | 0.3549 | 21.9712 | 4.1034 | 0.3631 | 21.7595 | 4.1323 | 0.3657 | 21.5684 | 3.9765 | 0.3519 | 20.7731 |
| | IPCC | 4.7661 | 0.4218 | 30.3209 | 4.4244 | 0.3915 | 23.2893 | 4.4221 | 0.3913 | 23.3136 | 4.0110 | 0.3550 | 20.8456 |
| | PNCF | 4.7203 | 0.4177 | 24.1462 | 4.6581 | 0.4122 | 21.5097 | 4.5503 | 0.4027 | 20.4041 | 4.5633 | 0.4038 | 19.6659 |
| | WSPred | 4.3792 | 0.3875 | 23.6124 | 4.1666 | 0.3687 | 22.3653 | 4.1252 | 0.3651 | 22.0316 | 4.0878 | 0.3618 | 22.1613 |
| | PLMF | 4.3158 | 0.3819 | 25.6351 | 4.1234 | 0.3649 | 24.5232 | 4.0839 | 0.3614 | 23.8452 | 4.0320 | 0.3568 | 22.1021 |
| | RTF | 4.1393 | 0.3663 | 22.7817 | 4.0253 | 0.3562 | 20.8091 | 3.8740 | 0.3428 | 20.1664 | 3.7992 | 0.3362 | 19.8977 |
| | TUIPCC | 4.0752 | 0.3606 | 22.5969 | 4.0366 | 0.3572 | 20.8590 | 3.8750 | 0.3429 | 20.1663 | 3.7076 | 0.3281 | 19.0516 |
| | RNCF | 4.2877 | 0.3794 | 23.1256 | 3.8378 | 0.3396 | 20.8402 | 4.2737 | 0.3782 | 19.7895 | 3.6536 | 0.3233 | 19.3801 |
| | GAFC | 4.0018 | 0.3541 | 21.9519 | 3.9006 | 0.3452 | 19.6847 | 3.7405 | 0.3310 | 19.2478 | 3.5385 | 0.3131 | 17.0031 |
| | DGNCL | 3.9824 | 0.3524 | 19.6318 | 3.9471 | 0.3493 | 18.3619 | 3.7739 | 0.3339 | 18.0271 | 3.6357 | 0.3217 | 17.5440 |
| | GACL | **3.6445** | **0.3225** | **14.1097** | **3.4006** | **0.3009** | **13.3413** | **2.8283** | **0.2503** | **12.4121** | **2.7658** | **0.2448** | **12.2824** |
| | Gains | 8.93% | 8.92% | 35.72% | 11.39% | 11.40% | 32.23% | 24.39% | 24.38% | 35.51% | 21.84% | 21.81% | 27.76% |

the best performance on both the RT and TP datasets. GAFC frequently achieves the second-best results across multiple metrics. Specifically, on the RT dataset, GACL achieved improvements ranging from 14.59% (MAE improvement compared to GAFC at a density of 20%) to 38.80% (RMSE improvement compared to PLMF at a density of 20%) over the best-performing baseline. On the TP dataset, GACL achieved improvements ranging from 8.92% (NMAE improvement compared to GAFC at a density of 5%) to 35.72% (RMSE improvement compared to GAFC at a density of 5%).

Notably, GACL's improvement in the RMSE metric is generally more significant than in MAE and NMAE, indicating better robustness against outliers compared to all baselines. This significant performance enhancement can be attributed to several key differences between GACL and GAFC. GAFC employs an enhanced GRU (EGRU) and a generative adversarial training mechanism for feature compensation, showing strong performance through temporal pattern extraction and feature enrichment via adversarial learning. However, while GAFC focuses on compensating feature losses in sequential modeling, GACL's target-prompt GAT effectively considers the complex implicit collaborative correlations and indirect invocation relationships between targets and their neighbors for every specific invocation. This allows GACL to extract the invocation-specific features of target users and services in each time slice and leverage the Transformer's powerful sequence modeling capabilities to learn the temporal evolution patterns of user and service features, achieving more precise QoS prediction than GAFC's EGRU-based approach.

Regarding the performance changes under different data density settings, we observe that the performance of GACL consistently outperforms the baselines. This indicates that by designing a target-prompt GAT and adopting multi-hop biased message passing, GACL effectively aggregates information from both directly and indirectly interacting neighbors to enhance the feature representation of target users and services, thereby alleviating the problem of accurately predicting QoS in sparse scenarios. In contrast, while GAFC attempts to address data sparsity through feature compensation, it lacks the explicit modeling of collaborative relationships in the user-service invocation graph that GACL employs. Additionally, as data density increases, the performance of all models improves; however, the gain in performance for GACL is more significant compared to the baselines. This is reflected in how GACL's relative gains expand with increased data density, demonstrating that GACL can more effectively extract valuable information from increasing data to achieve accurate temporal QoS prediction and integrate it into high-order temporal latent features for users and services.

Moreover, as shown in the table, GACL achieves significant improvements in QoS prediction performance across all experimental settings compared to our previous work, DGNCL. It demonstrates that the target-prompt GAT in GACL effectively identifies similar neighboring users and services under the guidance of target prompts. By simultaneously integrating historical QoS records, it facilitates high-quality deep feature extraction for users and services. Additionally, GACL enhances temporal feature extraction by replacing DGNCL's GRU with a Transformer encoder. The Transformer's self-attention mechanism enables comprehensive information aggregation across all time steps, addressing GRU's limitations in processing long-range dependencies. This architectural advancement allows GACL to more effectively capture global patterns and multi-scale temporal dynamics in QoS data.

TABLE IV
INFERENCE LATENCY COMPARISON FOR DIFFERENT METHODS ON USER-SERVICE INVOCATION PREDICTION

| Method | Category | Inference Latency (ms) |
|---|---|---|
| UPCC | CF-based | 3.2 |
| IPCC | CF-based | 3.5 |
| WSPred | MF-based | 5.8 |
| PNCF | NN-based | 8.4 |
| RTF | RNN-based | 15.3 |
| TUIPCC | CF-based | 7.6 |
| RNCF | RNN-based | 14.1 |
| GAFC | GAN+RNN-based | 20.5 |
| DGNCL | GNN+RNN-based | 21.8 |
| GACL | GNN+Transformer-based | 25.6 |

Overall, our proposed GACL achieves state-of-the-art performance on both the RT and TP datasets. Its excellent performance across different QoS density settings validates its effectiveness in handling QoS sparsity and capturing temporal features. The proposed target-prompt GAT and multi-layer Transformer encoder show significant advantages in high-order latent feature extraction and temporal evolution pattern mining for every specific target invocation, providing the robustness for precise temporal QoS prediction.

### D. Computational Efficiency Analysis

Beyond prediction accuracy, computational efficiency constitutes a critical criterion for real-world deployment of QoS prediction methods. We conducted a comprehensive analysis of inference latency across all competing methods for predicting single user-service pair QoS values, with results presented in Table IV.

As evidenced in Table IV, methods exhibit a clear complexity-latency relationship. Traditional collaborative filtering methods (UPCC, IPCC) demonstrate minimal latency (3.2-3.5 ms) due to their computational simplicity. Matrix factorization (WSPred) and neural network approaches (PNCF) show moderate latency (5.8-8.4 ms). Temporal modeling methods employing recurrent neural networks (RTF, RNCF) incur increased latency (14.1-15.3 ms) attributable to their sequential processing architecture. More sophisticated architectures incorporating multiple neural components exhibit progressively higher latency: GAFC (GAN+RNN, 20.5 ms), DGNCL (GNN+RNN, 21.8 ms), and our proposed GACL (GNN+Transformer, 25.6 ms).

This latency analysis reveals a computational efficiency-prediction accuracy trade-off inherent in QoS prediction models. While GACL requires additional computational resources, its 25.6 ms inference time remains well within practical bounds for real-time applications, particularly considering typical service response times range from hundreds of milliseconds to seconds. Furthermore, GACL's significant predictive improvements (14.59%-38.80% on RT dataset and 8.92%-35.72% on TP dataset versus the best baseline) justify this modest computational overhead. The target-prompt attention mechanism coupled with Transformer-based temporal modeling captures complex user-service interactions and temporal dependencies that simpler models cannot effectively represent. For latency-critical applications, techniques such as model quantization and hardware acceleration present viable optimization pathways without compromising prediction accuracy.

### E. Ablation Study

To evaluate the efficacy of our target-prompt attention strategy, we designed ablation experiments isolating its two key components: implicit collaborative relationships and historical QoS values. Experiments were conducted on RT and TP datasets with performance measured by NMAE and RMSE metrics. We designed three ablation variants:

- GACL-t: This variant excludes implicit collaborative relationship modeling between target and neighboring entities. Equation (5) becomes:

$$\hat{\mathbf{x}}_{s'}^{l-1} = norm\left(W_w^l w_{us'}^t + b_w^l\right) \tag{33}$$

- GACL-w: This variant omits historical QoS values while preserving implicit collaborative relationships. Consequently, eq. (5) becomes:

$$\hat{\mathbf{x}}_{s'}^{l-1} = norm\left(\mathbf{x}_s^{l-1} + \mathbf{x}_{s'}^{l-1}\right) \tag{34}$$

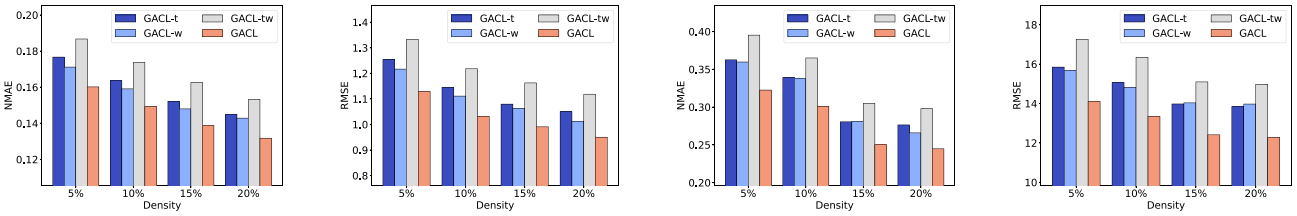
- GACL-tw: This variant eliminates both components, using only semantic relevance between target entities and their direct neighbors:

$$\left(\hat{attn}_{u\leftarrow s'}^t\right)^l = \left(attn_{u\leftarrow s'}^t\right)^l \tag{35}$$

Figure 3 shows that across all densities, the complete GACL consistently achieves the lowest error metrics, while GACL-tw exhibits the highest values, confirming the importance of the target-prompt module. The GACL-w and GACL-t variants demonstrate dataset-specific performance patterns: on RT data, GACL-w generally outperforms GACL-t, while on TP data, both variants show comparable performance.

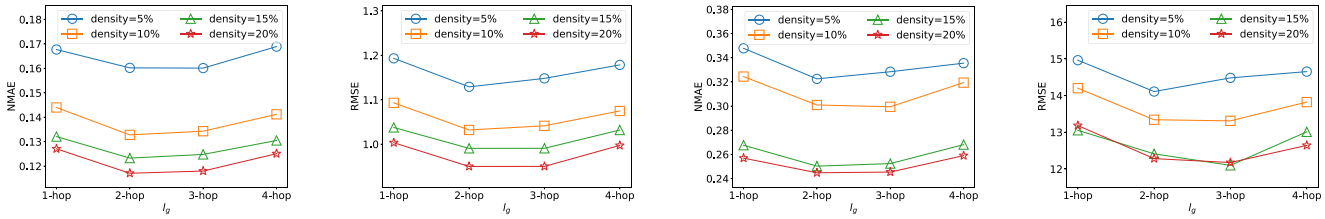
This discrepancy is likely related to the different historical QoS distributions of the two datasets. The QoS value range of RT is much smaller compared to TP, with fewer outliers. For a specific user-service invocation, its QoS value and implicit relationship with the network context of target users and services can be more effectively mined by deep models. Therefore, in the target-prompt module, considering only the implicit correlation between target users/services and neighbors can achieve good QoS prediction accuracy. Conversely, for the TP dataset with a wider distribution range and more outliers, a specific user-service invocation's QoS value does not always consistently correlate implicitly with the network context of target users/services. Thus, it requires additional consideration of the impact of historical QoS values. As a result, both GACL-t and GACL-w experience significant performance losses compared to GACL.

In summary, the ablation studies demonstrate the effectiveness of the target-prompt attention strategy. Learning context-aware features of users and services for each distinct invocation by simultaneously considering implicit collaborative relationships, and historical QoS values significantly enhances the model's predictive accuracy and robustness.

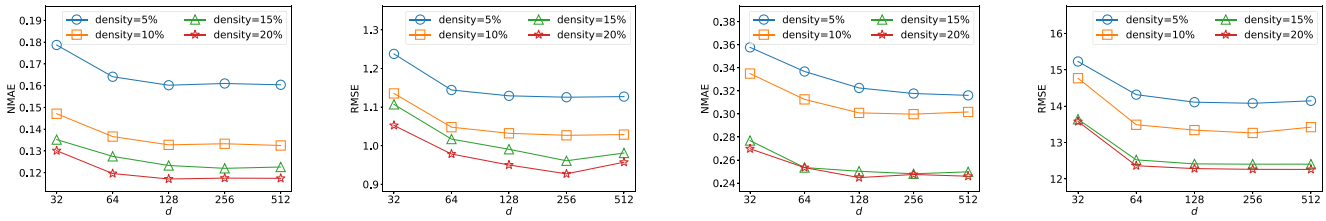



(a) NMAE of ablated variants on RT (b) RMSE of ablated variants on RT (c) NMAE of ablated variants on TP (d) RMSE of ablated variants on TP

Fig. 3. Ablation experiment results of NMAE and RMSE over RT and TP.




(a) NMAE of various $l_g$ on RT (b) RMSE of various $l_g$ on RT (c) NMAE of various $l_g$ on TP (d) RMSE of various $l_g$ on TP




(e) NMAE of various $d$ on RT (f) RMSE of various $d$ on RT (g) NMAE of various $d$ on TP (h) RMSE of various $d$ on TP

Fig. 4. Parameter impact of layers of target-prompt graph attention network $l_g$ and dimension of user/service feature $d$.

The superior performance of the original GACL across all metrics and datasets highlights the value of integrating these components in temporal QoS prediction.

### F. Performance Impact of Parameters

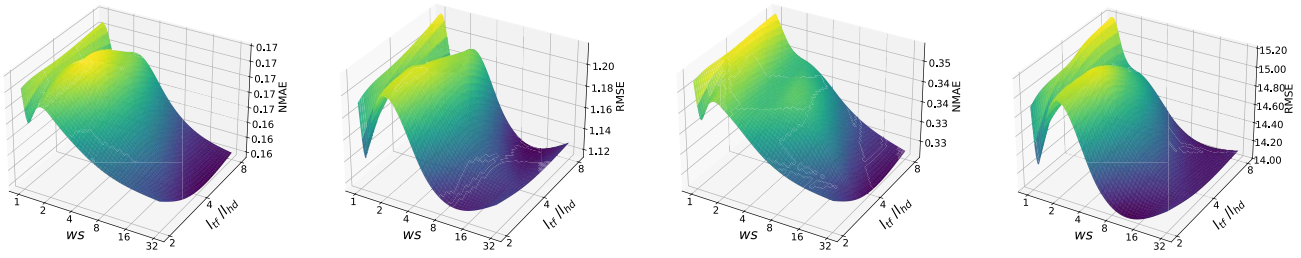
We investigate five key hyperparameters that impact our model's performance: layers of target-prompt graph attention network $(l_g)$, dimension of user/service features $(d)$, window size $(ws)$, transformer encoder layers $(l_{tf})$, and multi-head attention heads $(l_{hd})$. Through extensive experiments on RT and TP datasets across different density settings, we analyze how each parameter affects the model's feature learning and prediction capabilities.

*1) Impact of Target-Prompt GAT Layers $(l_g)$:* The target-prompt graph attention network layers $(l_g)$ directly influence the neighborhood scope considered during feature extraction, making it essential to validate its robustness across different data densities. We systematically evaluated the impact of varying $l_g$ from 1 to 4 across all density settings (5%-20%) on both RT and TP datasets, as illustrated in Figure 4a-4d. This comprehensive evaluation enables us to assess whether the model maintains stable prediction accuracy regardless of application scenario variations, thus confirming the model's practical robustness.

Experimental results demonstrate GACL consistently achieves optimal performance at $l_g = 2$ or $l_g = 3$ across all densities,with minimal performance fluctuations between these values. This stability stems from the target-prompt attention mechanism's adaptive calibration of neighborhood information based on contextual relevance. At $l_g = 1$, performance degrades significantly due to insufficient capture of indirect relationships through intermediate nodes. Conversely, at $l_g = 4$, over-smoothing occurs as excessive message passing dilutes node-specific features and introduces noise from irrelevant distant neighbors. For practical deployments, we recommend $l_g = 2$ for resource-constrained environments and $l_g = 3$ when prioritizing prediction accuracy. This parameter's robustness across density settings significantly simplifies model deployment in diverse service environments while maintaining high prediction performance.

*2) Impact of Feature Dimension $(d)$:* Feature dimension $(d)$ significantly impacts representation capacity in deep learning models. To thoroughly evaluate dimension robustness, we conducted extensive experiments across all density settings (5%-20%) on both RT and TP datasets, systematically varying $d$ from 32 to 512, as shown in Figure 4e-4h. This comprehensive analysis reveals how feature dimension affects model stability and prediction accuracy in diverse application scenarios, a critical factor for practical deployment.

GACL maintains stable performance within the dimension range of 128-256 across all density settings. This dimensional robustness stems from the synergistic interaction between the target-prompt attention mechanism, which calibrates feature importance based on prediction context, and the Transformer's multi-head attention, which captures diverse




(a) NMAE of various $ws$ & $l_{tf}$ & $l_{hd}$ on RT

(b) RMSE of various $ws$ & $l_{tf}$ & $l_{hd}$ on RT

(c) NMAE of various $ws$ & $l_{tf}$ & $l_{hd}$ on TP

(d) RMSE of various $ws$ & $l_{tf}$ & $l_{hd}$ on TP

Fig. 5. Parameter impact of window size $ws$, transformer encoder layers $l_{tf}$, and attention heads $l_{hd}$.

temporal patterns in parallel. At lower dimensions (32-64), performance degrades significantly due to insufficient representational capacity, particularly in higher density settings. Conversely, large dimensions (>256) introduce excessive parameters without proportional accuracy gains, increasing computational overhead while yielding minimal improvement. We recommend $d = 128$ for resource-constrained environments and $d = 256$ when maximum prediction precision is required. This dimensional stability facilitates deployment across diverse computational environments without sacrificing prediction accuracy.

*3) Impact of Window Size and Transformer Parameters ($ws$, $l_{tf}$, $l_{hd}$):* Figure 5a-5d illustrates the interdependence between window size $(ws)$, transformer encoder layers $(l_{tf})$, and attention heads $(l_{hd})$. Following established convention [21], we maintain $l_{hd} = l_{tf}$ to ensure balanced representational capacity.

We observe that performance improves with increasing $ws$ and $l_{tf}$ (and consequently $l_{hd}$). Importantly, the optimal configuration depends on their interaction: smaller $ws$ values work best with fewer transformer layers and attention heads, while larger $ws$ values require more layers and heads. This pattern emerges because smaller temporal windows focus on short-term dependencies that can be modeled with simpler architectures, while larger windows capture long-term patterns requiring more complex models. The increased attention heads enable parallel processing of different temporal aspects, while deeper transformer layers facilitate more sophisticated feature transformations.

Notably, while the leading baselines GAFC and our previous DGNCL model (both utilizing GRU-based temporal modeling) exhibit performance degradation beyond specific window thresholds ($ws \geq 10$ for GAFC [20]; $ws \geq 25$ for DGNCL [19]), our Transformer-based GACL maintains robust performance with window sizes up to 32. This superior capability derives from several architectural advantages: (1) the self-attention mechanism directly models dependencies between arbitrary time points regardless of temporal distance, circumventing the sequential compression limitations of RNNs; (2) parallel processing enables simultaneous consideration of all historical time slices; and (3) multi-head attention concurrently captures diverse aspects of temporal patterns, effectively modeling the multi-dimensional nature of QoS variations.

Based on comprehensive parameter analysis, we selected the configuration $ws = 32$, $l_{tf} = l_{hd} = 8$ for optimal balance between prediction accuracy and computational efficiency.

## V. Discussion

### A. Prediction Accuracy and Practical Utility

Beyond GACL's consistent performance advantage over existing approaches, an assessment of its absolute prediction errors confirms practical utility in real-world QoS prediction contexts. The RT dataset results (density = 20%) reveal an RMSE of 0.9500 seconds—representing merely 4.75% of the value range (0-20 seconds) and substantially lower than the standard deviation (6.1 seconds). Similarly, with the TP dataset, the RMSE of 12.2824 kbps constitutes only 0.18% of the throughput range (0-6727 kbps), well below the standard deviation (54.3 kbps). These metrics demonstrate GACL's significant potential for deployment in operational service environments.

In service-oriented computing, the critical applications of QoS prediction—service selection and recommendation—primarily require accurate service ranking rather than precise value estimation [30]. GACL's achieved accuracy levels preserve the correct ranking order, enabling optimal service selection decisions. This capability proves particularly valuable in large-scale ecosystems where users must differentiate among functionally equivalent services based solely on QoS properties. The prediction errors attained by GACL match or exceed the benchmarks established in prior research, validating its effectiveness for practical service selection applications.

### B. Generalizability Analysis

Although initially developed for QoS prediction on the WS-DREAM dataset, GACL's Target-prompt graph attention mechanism addresses fundamental challenges transcending specific datasets through its core principle—calculating attention weights by integrating implicit collaborative relationships with historical QoS values. Ablation studies confirm these components' effectiveness in extracting meaningful patterns from user-service interaction data. The underlying methodology extends to any domain where entity interactions exhibit temporal evolution patterns within graph structures. GACL's key strength lies in its adaptive capability to automatically calibrate the relative importance of structural and

temporal information based on contextual factors and data characteristics.

This generalizability stems from three critical architectural decisions: (1) The Target-prompt mechanism offers a generalizable approach to context-aware graph learning, addressing the universal challenge of balancing diverse information sources in graph-based prediction tasks; (2) The Transformer-based temporal feature extraction module captures complex non-linear dependencies without domain-specific assumptions; and (3) The hierarchical representation learning framework accommodates varying interaction complexities. These design elements extend GACL's applicability beyond QoS prediction to recommendation systems, traffic flow prediction, and social network analysis—domains where entity relationships form graph structures and temporal dynamics influence predictions.

### C. Robustness Analysis

The WS-DREAM dataset, despite being singular, provides substantial advantages for robustness assessment. Unlike synthetic alternatives, it features heterogeneous QoS distributions with distinct statistical properties between RT and TP metrics, significant regional variations across its global scope (57 service regions, 22 user countries), and temporal fluctuations throughout 64 time slices. These inherent characteristics create a comprehensive testbed for model robustness by simulating the distribution shifts encountered in operational environments. The geographical diversity particularly challenges model stability, as QoS values between distant users and services exhibit fundamentally different network characteristics compared to proximate ones, necessitating adaptive feature extraction.

From a theoretical perspective, GACL exhibits exceptional robustness against distribution variations through its architectural principles. The target-prompt attention mechanism employs affine transformation, enabling dynamic calibration of neighborhood information contribution without assuming specific distribution patterns. This self-adjusting property facilitates adaptation to diverse QoS distribution characteristics without parameter modifications. Empirical evaluations across varying data density settings (5%-20%) confirm stability under different sparsity conditions, with consistent performance advantages over baselines. This stability derives from adaptive normalization and calibrated attention mechanisms that prevent overreliance on either collaborative or historical information sources.

## VI. Related Work

### A. Static QoS Prediction

Static QoS prediction approaches can be classified into three categories: memory-based, model-based, and deep learning-based. These methods typically operate on a two-dimensional matrix representing user-service QoS invocations.

Memory-based methods primarily utilize traditional collaborative filtering (CF) techniques to predict missing QoS values. These approaches can be further divided into user-based [2], service-based [3], and hybrid methods that combine both user-based and service-based predictions using weighted coefficients. The fundamental principle of memory-based QoS prediction is to identify a set of similar users or services (neighborhood) through similarity calculations and then use this neighborhood to compute deviation migrations. These migrations are combined with average QoS values to predict the missing QoS. Wu et al. [7] introduced a rate-based similarity (RBS) method to select user and service neighborhoods, achieving better QoS predictions. Zou et al. [8] proposed a reinforced CF approach that combines RBS and Pearson Correlation Coefficient (PCC) to accurately calculate average QoS values and deviation migrations.

Model-based approaches aim to extract implicit linear or nonlinear invocation relationships to enhance QoS prediction performance, partially addressing the limitations of CF-based methods. Xu et al. [31] proposed two context-aware matrix factorization models that improve QoS prediction by considering user and service contexts. Wu et al. [32] introduced a general context-sensitive matrix factorization approach to model interactions between users and services more effectively.

Deep learning techniques have recently been employed to solve QoS prediction problems due to their ability to handle data sparsity and learn implicit nonlinear interactions. These methods often combine neural networks with matrix factorization and adopt multi-task learning to reduce prediction errors and improve performance. For instance, Xu et al. [33] developed the model that integrate deep learning with matrix factorization to enhance prediction accuracy. Li et al. [18] proposed the topology-aware neural (TAN) model, which considers the underlying network topology and complex interactions between autonomous systems to improve collaborative QoS prediction. Zou et al. [34] designed a location-aware two-tower deep residual network combined with collaborative filtering, achieving superior QoS prediction. Recent advancements have further improved QoS prediction performance by incorporating expert systems and attention mechanisms [35], or using GNNs [36], [37] to select, extract, and interact with multiple features from user-service contextual information and QoS invocations.

### B. Temporal QoS Prediction

Temporal QoS prediction can be partitioned into four categories: temporal factor integrated CF, sequence prediction, tensor decomposition, and deep learning.

Temporal factor integrated CF methods incorporate temporal information into the collaborative filtering process. Hu et al. [9] integrated temporal factors with the CF approach and used a random walk algorithm to select more similar neighbors, alleviating data sparsity and improving temporal QoS prediction. Tong et al. [11] improved temporal QoS prediction by normalizing historical QoS values, calculating similarity, and selecting neighbors based on the distance of time slices, then using hybrid CF for prediction. These approaches demonstrate the effectiveness of integrating temporal information into QoS prediction and addressing the limitations of non-temporal QoS prediction approaches.

Sequence prediction methods use time series analysis techniques to enhance temporal QoS prediction. Hu et al. [12] combined CF with the ARIMA model and applied the Kalman filtering algorithm to compensate for ARIMA's shortcomings in temporal QoS prediction. Ding et al. [13] integrated the ARIMA model with memory-based CF to capture the temporal characteristics of user similarity, improving the accuracy of missing temporal QoS predictions. These approaches highlight the benefits of combining sequence prediction analysis with QoS prediction to capture temporal characteristics of QoS.

Tensor decomposition methods convert the classic two-dimensional user-service matrix into a three-dimensional tensor representation, enabling temporal factor incorporation. Meng et al. [14] proposed a temporal hybrid collaborative cloud service recommendation approach using CP decomposition and a biases model to distinguish temporal QoS metrics from stable ones. Zhang et al. [16] combined Personalized Gated Recurrent Unit and Generalized Tensor Factorization to leverage long-term dependency patterns for comprehensive temporal QoS prediction. These methods show the effectiveness of using tensor representations and factorization techniques to incorporate temporal factors into QoS prediction.

Deep learning models, such as RNN and its variants LSTM and GRU, have been increasingly used for temporal QoS prediction. Xiong et al. [15] proposed a personalized matrix factorization approach based on LSTM to capture dynamic representations for online QoS prediction. Zou et al. [5] developed a temporal QoS prediction framework that combines binary features with memory-based similarity and feeds them to a GRU model to mine temporal aggregated features for predicting unknown temporal QoS values. These deep learning approaches effectively capture temporal dependencies and patterns, enhancing the accuracy of temporal QoS prediction.

## VII. Conclusion and Future Work

In this paper, we propose a novel framework for temporal QoS prediction, named **G**raph **A**ttention **C**ollaborative **L**earning (**GACL**). Specifically, we first leverage a dynamic user-service invocation graph to model historical interactions. Then, we design a target-prompt graph attention network to extract invocation-specific deep latent features of users and services. The target-prompt attention mechanism enhances feature extraction by considering both implicit collaborative relationships between neighbors and target users and services, and historical QoS values of corresponding user-service invocations. This dual consideration enables our model to adaptively calibrate attention weights for each specific invocation context, significantly improving feature representation quality in sparse service environments. Finally, the multi-layer Transformer encoder further uncovers feature temporal evolution patterns for users and services, providing a comprehensive solution for accurate QoS prediction. Extensive experiments on the WS-DREAM dataset demonstrate GACL's superiority over state-of-the-art methods, confirming the effectiveness of our framework for accurate temporal QoS prediction.

In the future, we plan to focus on enhancing the GACL framework, which includes optimizing the architecture for various service ecosystems and integrating additional contextual information. We also aim to explore the framework's generalizability to more complex and dynamic service environments with heterogeneous QoS distributions and varying sparsity patterns, ensuring its practical applicability across diverse real-world deployment scenarios.

## References


[1] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world Web services," *IEEE Trans. Services Comput.*, vol. 7, no. 1, pp. 32–39, Jan./Mar. 2012.

[2] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for Web services via collaborative filtering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2007, pp. 439–446.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.

[4] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware Personalized QoS prediction framework for Web services," in *Proc. Int. Symp. Softw. Rel. Eng. (ISSRE)*, 2011, pp. 210–219.

[5] G. Zou et al., "DeepTSQP: Temporal-aware service QoS prediction via deep neural network and feature integration," *Knowl.-Based Syst.*, vol. 241, pp. 108062–108062, Apr. 2022.

[6] W. Liang, Y. Li, J. Xu, Z. Qin, D. Zhang, and K.-C. Li, "QoS prediction and adversarial attack protection for distributed services under DLaaS," *IEEE Trans. Comput.*, vol. 73, no. 3, pp. 669–682, Mar. 2024.

[7] X. Wu, B. Cheng, and J. Chen, "Collaborative filtering service recommendation based on a novel similarity computation method," *IEEE Trans. Services Comput.*, vol. 10, no. 3, pp. 352–365, May/Jun. 2015.

[8] G. Zou, M. Jiang, S. Niu, H. Wu, S. Pang, and Y. Gan, "QoS-aware Web service recommendation with reinforced collaborative filtering," in *Proc. Int. Conf. Service-Oriented Comput. (ICSOC)*, 2018, pp. 430–445.

[9] Y. Hu, Q. Peng, and X. Hu, "A time-aware and data sparsity tolerant approach for Web service recommendation," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2014, pp. 33–40.

[10] H. Ma, H. Zhu, Z. Hu, W. Tang, and P. Dong, "Multi-valued collaborative QoS prediction for cloud service via time series analysis," *Future Gener. Comput. Syst.*, vol. 68, pp. 275–288, Mar. 2017.

[11] E. Tong, W. Niu, and J. Liu, "A missing QoS prediction approach via time-aware collaborative filtering," *IEEE Trans. Services Comput.*, vol. 15, no. 6, pp. 3115–3128, Nov./Dec. 2021.

[12] Y. Hu, Q. Peng, X. Hu, and R. Yang, "Web service recommendation based on time series forecasting and collaborative filtering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2015, pp. 233–240.

[13] S. Ding, Y. Li, D. Wu, Y. Zhang, and S. Yang, "Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model," *Decis. Support Syst.*, vol. 107, pp. 103–115, Mar. 2018.

[14] S. Meng et al., "A temporal-aware hybrid collaborative recommendation method for cloud service," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2016, pp. 252–259.

[15] R. Xiong, J. Wang, Z. Li, B. Li, and P. C. Hung, "Personalized LSTM based matrix factorization for online QoS prediction," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2018, pp. 34–41.

[16] Y. Zhang, C. Yin, Z. Lu, D. Yan, M. Qiu, and Q. Tang, "Recurrent tensor factorization for time-aware service recommendation," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105762.

[17] P. Zhang, W. Huang, Y. Chen, and M. Zhou, "Predicting quality of services based on a two-stream deep learning model with user and service graphs," *IEEE Trans. Services Comput.*, vol. 16, no. 6, pp. 4060–4072, Nov./Dec. 2023.

[18] J. Li, H. Wu, J. Chen, Q. He, and C.-H. Hsu, "Topology-aware neural model for highly accurate QoS prediction," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 7, pp. 1538–1552, Jul. 2022.

[19] S. Hu et al., "Temporal-aware QoS prediction via dynamic graph neural collaborative learning," in *Proc. Int. Conf. Service-Oriented Comput. (ICSOC)*, 2022, pp. 125–133.

[20] P. Zhang, Y. Chen, W. Huang, H. Zhu, and Q. Zhao, "Generative-adversarial-based feature compensation to predict quality of service," *IEEE Trans. Services Comput.*, vol. 17, no. 1, pp. 209–223, Jan./Feb. 2024.

[21] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 1–11.
[22] J. T. Aparício, E. Arsenio, F. Santos, and R. Henriques, "Using dynamic knowledge graphs to detect emerging communities of knowledge," *Knowl. Based Syst.*, vol. 294, pp. 111671–111671, Jun. 2024.
[23] J. Nie, X. Wang, R. Hou, G. Li, H. Chen, and W. Zhu, "Dynamic Spatio-temporal graph reasoning for VideoQA with self-supervised event recognition," *IEEE Trans. Image Process.*, vol. 33, pp. 4145–4158, 2024.
[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on Imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1026–1034.
[27] I. Loshchilov and F. Hutter, "Decoupled weight decay Regularization," 2017, *arXiv:1711.05101*.
[28] T. Liang, M. Chen, Y. Yin, L. Zhou, and H. Ying, "Recurrent neural network based collaborative filtering for QoS prediction in IoV," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2400–2410, Mar. 2022.
[29] L. Chen, A. Zheng, Y. Feng, F. Xie, and Z. Zheng, "Software service recommendation based on collaborative filtering neural network model," in *Proc. Int. Conf. Service-Oriented Comput. (ICSOC)*, vol. 16, 2018, pp. 388–403.
[30] Y. Syu, C. Wang, and Y. Fanjiang, "A survey of time-aware dynamic QoS forecasting research, its future challenges and research directions," in *Proc. Int. Conf. Services Comput. (SCC)*, 2018, pp. 35–49.
[31] Y. Xu, J. Yin, S. Deng, N. N. Xiong, and J. Huang, "Context-aware QoS prediction for Web service recommendation and selection," *Expert Syst. Appl.*, vol. 53, pp. 75–86, Jul. 2016.
[32] H. Wu, K. Yue, B. Li, B. Zhang, and C.-H. Hsu, "Collaborative QoS prediction with context-sensitive matrix factorization," *Future Gener. Comput. Syst.*, vol. 82, pp. 669–678, May 2018.
[33] J. Xu et al., "NFMF: Neural fusion matrix factorisation for QoS prediction in service selection," *Connect. Sci.*, vol. 33, no. 3, pp. 753–768, 2021.
[34] G. Zou et al., "NCRL: Neighborhood-based collaborative residual learning for adaptive QoS prediction," *IEEE Trans. Services Comput.*, vol. 16, no. 3, pp. 2030–2043, May/Jun. 2023.
[35] H. Lian, J. Li, H. Wu, Y. Zhao, L. Zhang, and X. Wang, "Toward effective personalized service QoS prediction from the perspective of multi-task learning," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 2587–2597, Sep. 2023.
[36] Z. Wu, D. Ding, Y. Xiu, Y. Zhao, and J. Hong, "Robust QoS prediction based on reputation integrated graph convolution network," *IEEE Trans. Services Comput.*, vol. 17, no. 3, pp. 1154–1167, May/Jun. 2024.
[37] F. Bi, T. He, Y. Xie, and X. Luo, "Two-stream graph convolutional network-incorporated latent feature analysis," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 3027–3042, Jul./Aug. 2023.

**Shengxiang Hu** received the master's degree in computer science and technology from Shanghai University, China, in 2021, where he is currently pursuing the Ph.D. degree with the School of Computer Engineering and Science. Over the course of his academic career, he has contributed significantly to the field through his authorship and co-authorship of 15 scholarly papers. These papers have been published in esteemed international journals and presented at prestigious conferences, such as *Knowledge-Based Systems*, IEEE TRANSACTIONS ON SERVICE COMPUTING, AAAI, ICSOC, and PPSN. His primary areas of research encompass quality of service prediction, graph neural networks, and natural language processing.

**Guobing Zou** received the Ph.D. degree in computer science from Tongji University, Shanghai, China, 2012. He is a Full Professor and the Vice Dean of the School of Computer Science, Shanghai University, China. He has worked as a Visiting Scholar with the Department of Computer Science and Engineering, Washington University, St. Louis, from 2009 to 2011, USA. He has published more than 120 papers on premier international journals and conferences, including IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE ICWS, ICSOC, IEEE SCC, AAAI, *Information Sciences*, *Expert Systems With Applications*, and *Knowledge-Based Systems*. His current research interests mainly focus on services computing, edge computing, data mining and intelligent algorithms, and recommender systems.

**Bofeng Zhang** received the Ph.D. degree from Northwestern Polytechnic University, China, in 1997. He is a Full Professor and the Dean of the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China. He experienced a Postdoctoral Research with Zhejiang University, China, from 1997 to 1999. He worked as a Visiting Professor with the University of Aizu, Japan, from 2006 to 2007. He worked as a Visiting Scholar with Purdue University, USA, from 2013 to 2014. He has published more than 200 papers on international journals and conferences. His research interests include personalized service recommendation, intelligent human-computer interaction, and data mining. He worked as the Program Chair for UUMA and ICSS. He also served as a program committee member for multiple international conferences.

**Shaogang Wu** received the bachelor's degree in computer science and technology from Shanghai University, China, in 2020, where he is currently pursuing the master's degree with the School of Computer Engineering and Science. He has published two papers on ICSOC 2022 and TSC. He has led research and development group to successfully design and implement a service-oriented enterprise application platform, which can intelligently classify and recycle, cultivate citizens' habit of throwing recyclables, and produce significant economic and social benefits by providing high QoS. His research interests include service quality management, deep learning, and intelligent algorithms.

**Shiyi Lin** received the bachelor's and master's degrees in computer science and technology from Shanghai University, in 2021 and 2024, respectively, where he is currently pursuing the Ph.D. degree with the School of Computer Engineering and Science. He has published one paper on International Conference on Service-Oriented Computing, and submitted two papers on International Conference on Web Services and IEEE TRANSACTIONS ON SERVICES COMPUTING. His research interests include graph representation learning, recommendation systems and service computing.

**Yanglan Gan** received the Ph.D. degree in computer science from Tongji University, Shanghai, China, 2012. She is a Full Professor with the School of Computer Science and Technology, Donghua University, Shanghai. She has published more than 50 papers on premier international journals and conferences, including *Bioinformatics*, *Briefings in Bioinformatics*, *BMC Bioinformatics*, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE ICWS, ICSOC, *Neurocomputing*, and *Knowledge-Based Systems*. Her research interests include bioinformatics, service computing, and data mining.

**Yixin Chen** (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign in 2005. He is currently a Full Professor of Computer Science with Washington University, St. Louis, MO, USA. He has published more than 210 papers on premier international journals and conferences, including *Artificial Intelligence*, *Journal of Artificial Intelligence Research*, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IJCAI, AAAI, ICML, and KDD. His research interests include artificial intelligence, data mining, deep learning, and big data analytics. He won the best paper award with AAAI and a best paper nomination at KDD. He received an Early Career Principal Investigator Award from the U.S. Department of Energy and a Microsoft Research New Faculty Fellowship. He was an Associate Editor for the *ACM Transactions on Intelligent Systems and Technology*, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and *Journal of Artificial Intelligence Research*.