

LGA: LLM-GNN Aggregation for Temporal Evolution Attribute Graph Prediction

Feng Zhao, Ruoyu Chai, Kangzheng Liu, Xianggan Liu*

Natural Language Processing and Knowledge Graph Lab,

School of Computer Science and Technology,

Huazhong University of Science and Technology, Wuhan, China

zhaof@hust.edu.cn, 1343125122@qq.com, {frankluis, liuxg}@hust.edu.cn

Abstract

Temporal evolution attribute graph prediction, a key task in graph machine learning, aims to forecast the dynamic evolution of node attributes over time. While recent advances in Large Language Models (LLMs) have enabled their use in enhancing node representations for integration with Graph Neural Networks (GNNs), their potential to directly perform GNN-like aggregation and interaction remains underexplored. Furthermore, traditional approaches to initializing attribute embeddings often disregard structural semantics, limiting the provision of rich prior knowledge to GNNs. Current methods also primarily focus on 1-hop neighborhood aggregation, lacking the capability to capture complex structural interactions. To address these limitations, we propose a novel prediction framework that integrates structural information into attribute embeddings through the introduction of an attribute embedding loss. We design specialized prompts to enable LLMs to perform GNN-like aggregation and incorporate a relation-aware Graph Convolutional Network to effectively capture long-range and complex structural dependencies. Extensive experiments on multiple real-world datasets validate the effectiveness of our approach, demonstrating significant improvements in predictive performance over existing methods.

1 Introduction

In real-world scenarios, it is essential to predict the temporal changes of a specific entity's attributes, such as fault detection, weather forecasting, and trade analysis. In most cases, entities exist within groups, and the interactions between individuals and groups serve as key factors in understanding the dynamics of attributes. These scenarios are thus often modeled as temporal evolution attribute graphs. For instance, in a trade graph,

nodes represent countries, while attributes correspond to the inherent assets of those countries. Research on temporal evolution attribute graph prediction has evolved from data-level approaches (Liu et al., 2011; Jin et al., 2023; Salamanis et al., 2015; Flunkert et al., 2017) to graph-structured methodologies (Jin et al., 2020), with methods transitioning from traditional graph neural networks (GNNs) to hybrid techniques that leverage large language models (LLMs) for enhanced performance.

As shown in Figure 1, recent methods treat LLMs and GNNs in isolation, applying them to separate stages of temporal evolution attribute graph prediction. LLMs are designed to enhance the initial node representations through attribute text, while GNNs capture the structural features of the graph. Although (Yang et al., 2021) and (Zhao et al., 2022) attempt to iteratively enhance each other by layering GNN and LLM, the two models still operate independently. This paper aims to explore whether it is possible to design a GNN-like aggregation method for LLMs, enabling simultaneous understanding of attributes and aggregation of graph structure within the LLM framework.

Additionally, recent methods generate initial attribute embeddings with weak semantic meaning, failing to account for the topological semantics inherent in the nodes. This neglect of the semantic content of attribute embeddings can make it difficult for subsequent modules, such as GNNs, to effectively recognize and differentiate between nodes. This issue becomes especially pronounced when node attributes are sparse or when there is low distinguishability between attributes, significantly increasing the difficulty for the model to understand nodes. For newly added nodes in the graph, which lack historical data, the model must rely on embeddings with strong representational power to inform link prediction and attribute change. Therefore, enhancing the semantic richness of initial attribute embeddings is crucial for improving the overall

*Co-first author

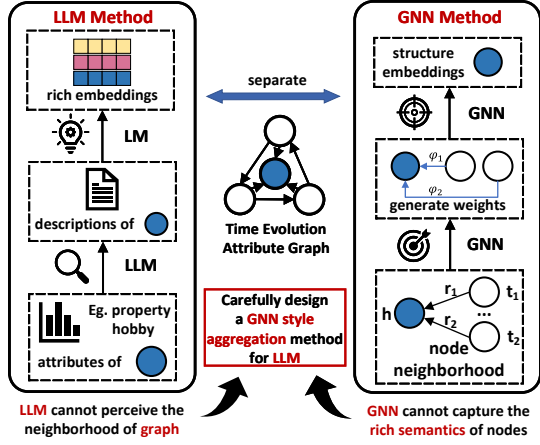


Figure 1: Isolation of LLM and GNN in temporal evolution attribute graph prediction

performance of the model, which is a key focus of this paper.

Moreover, recent approaches in the aggregation process are limited to considering only one-hop neighborhoods, lacking the ability to capture complex structural interactions beyond a limited hop. This limitation makes it difficult for the model to gather complete node interaction information, hindering its understanding of the graph structure. Moreover, any bias in this understanding can amplify errors in weight parameters, and in multi-step prediction scenarios, these errors will be further magnified, affecting the accuracy of attribute predictions.

We propose LGA, a LLM-GNN Aggregation model for temporal evolution attribute graph prediction. To implement a GNN-like aggregation method based on LLMs, we design prompts that describe node neighborhood information in the form of links while constraining the LLM to output results according to the priority of neighbors. To integrate node topological features into attribute embeddings, we introduce an attribute embedding loss based on cross-entropy to minimize the distance between the topological space and the pre-embedding space. Additionally, to capture complex structural interactions beyond limited hops, we incorporate a relation-aware Graph Convolutional Network (R-GCN), enabling long-range neighborhood aggregation through multi-layer information propagation. Our main contributions are as follows:

- We propose LGA, a LLM-GNN aggregation model for temporal evolution attribute graph prediction, aimed at enhancing the accuracy of numerical attribute prediction.

- We implemented a GNN-like aggregation process based on a LLM by designing tailored prompts, introduced an attribute embedding loss to address the issue of weak semantic representation in attribute embeddings, and incorporated an R-GCN module to capture long-range structural interactions between nodes, thereby reducing model interpretation bias.
- We evaluate the prediction accuracy of our model, LGA, using real-world datasets from five different domains. The results of multiple experiments confirm the effectiveness of our model over other approaches.

2 Related work

2.1 Attribute Sequence Prediction

Attribute sequence prediction relies on time series prediction methods to model and directly predict actual value data (attributes). With the development of deep learning, several approaches employing deep learning, such as GRU (Chung et al., 2014), the CNN-like WaveNet algorithm, the attention-based Transformer (Vaswani et al., 2017) and the Informer and TFT algorithms, have high learning freedom and can self-design loss functions for end-to-end training. These algorithms use attribute values to generate adaptive embeddings, learn temporal evolution trends, and finally decode the updated attribute embeddings and return prediction values.

2.2 Attribute Graph Prediction

The traditional prediction method based on graph sequence data directly concatenates entities and attributes, and connects them in the form of triplets (h, r, t) , ultimately decoding the attribute information. (Ma et al., 2019; Tresp et al., 2017; Han et al., 2021) achieve entity representation on dynamic knowledge graphs by studying temporal features. (Han et al., 2020) rigorously defines forecasting task on tKGs. For example, the diffusion convolutional RNN (DCRNN) model (Andreoletti et al., 2019) used bidirectional random walks and information decoding to obtain structural information and time information, respectively but it cannot be applied to dynamic graphs and multiple relationship graphs that evolve over time. As research has progressed, the RGNMF-AN model (Nasiri et al., 2023) was used to observe the hidden potential contained in attributes and model attribute information for direct graph link prediction. However, simultaneously predicting the topological and attribute

information of nodes remains a challenge. DART-NET (Garg et al., 2020) was used to conduct attribute mapping prediction at unknown time points, construct a new timestamp graph structure and perform attribute generation through operations such as average pooling and GRUs. An improved DeepWalk (Berahmand et al., 2021) algorithm was developed based on the idea that structural similarity and attribute similarity are completely independent issues. However, this approach neglects the semantic impacts of the same edges, resulting in attribute embedding and structural detachment. FC-STGNN (Wang et al., 2024) models and learns the correlation between attributes as additional information for prediction, expanding the model’s perspective and achieving considerable results.

2.3 Research on LLMs in Temporal Evolution Attribute Graphs

The current focus of work (Zhang et al., 2023) on expressing large language models in graph structures is mostly on traditional graph tasks such as link prediction, loop detection, and topology sorting. (You et al., 2024) pointed out that the existing fusion methods of LLM and GNN only integrate the two into different stages, such as (Huang et al., 2024). (Wang et al., 2023) proposed the NLGraph as a universal paradigm for solving graph tasks. However, the graph size is small and there is no research on temporal attribute prediction related issues. TAPE (He et al., 2023) studied the semantic enhancement of LLM on text attribute graphs and applied it in conjunction with GNN for node classification tasks. Although the effect is excellent, it is difficult to transfer to nontext attribute graphs. LLM4TS (Chang et al., 2023) studied the solving ability of LLM in multivariate time series prediction, but it was not conducted in the context of time series diagrams.

3 Methodology

In this section, we first define the symbols needed for to elaborate and describe the integral model. Then, we elaborate on the functional principles and implementation details of each module within our LGA model.

3.1 Overview of Symbols and Models

The temporal evolution attribute graph consists of a series of tuples, represented as $\mathcal{G}_\tau = \{(h, r, t, a_h^\tau, a_t^\tau, \tau)_1, \dots, (h, r, t, a_h^\tau, a_t^\tau, \tau)_n\}$, where h denotes the head node, t denotes the tail

node, r represents the relationship category, and τ represents the timestamp. a_h^τ and a_t^τ represent the attribute information of the head node and tail node, respectively. It is important to note that the attributes here are numerical rather than textual descriptions. In summary, G_τ represents the attribute graph at time τ consisting of n tuples, while the sequence of attribute graphs required for prediction is denoted as $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_\tau\}$. To learn the representations of the attribute graph-related information contained in the model, we define a learnable entity embedding matrix as E_τ and a relationship embedding matrix as R_τ . We initialize the embedding matrices for all nodes and relationship categories using Xavier Uniform, denoted as \mathbf{E}_τ and \mathbf{R}_τ , and update them continuously during the training process.

As depicted in Fig. 2, our LGA framework consists of three stages in sequence: the initialization embedding stage, the neighborhood aggregation stage, and the prediction stage. In the initialization embedding stage, we introduce a loss function to align the embedding space with the topological space, effectively reducing the distance between them. In the neighborhood aggregation stage, we deploy the R-GCN module to capture long-distance complex neighborhood interactions. Additionally, we design prompts that describe node neighborhoods in the form of links, guiding the LLM to generate aggregation-focused outputs, thereby enabling a GNN-like aggregation method using LLMs. Finally, the results from both aggregation modules are concatenated to form the complete aggregation sequence, which is then used by the GRU to capture temporal features.

3.2 LGA: Our Framework

3.2.1 Initialization Embedding Stage

To generate reasonable expressions for attribute embeddings, we generate attribute embeddings through fully connected layers to facilitate fusion and generate complete node information:

$$\mathbf{a}_h^\tau = \sigma(W_a \cdot a_h^\tau). \quad (1)$$

Here, $a_h^\tau \in \mathbb{R}^{v \times 1}$ denotes the attribute value of node h at timestamp τ , $\mathbf{a}_h^\tau \in \mathbb{R}^d$ represents the comprehensive embeddings of all the attributes, $W_a \in \mathbb{R}^{v \times d}$ denotes the learnable parameter matrix and $\sigma(\cdot)$ represents the non-linear rectified linear unit (ReLU).

We then add a loss function to narrow the attribute embeddings of the nodes that are close

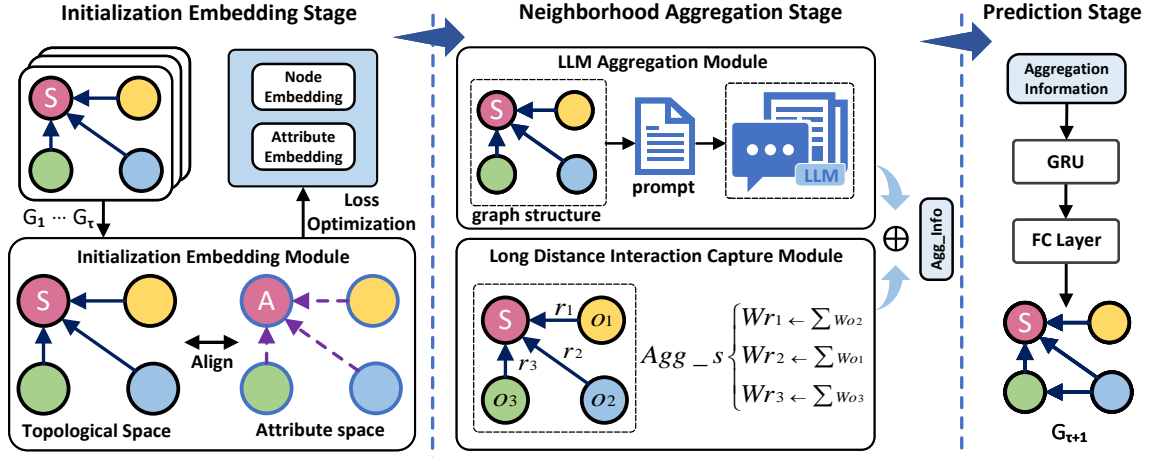


Figure 2: Framework of LGA. The framework consists of three stages: initialization embedding stage, neighborhood aggregation stage, and the prediction stage. The content below each stage represents the modules we have designed to overcome challenges.

to each other and to expand the embeddings of the nodes that are far away. Inspired by the DEAL (Hao et al., 2020) model, we modify and set the loss as follows:

$$\mathcal{L}_a = \frac{1}{|H|} \left(\sum_0^{|H_1|} \omega(h, t_k) \varphi(\cos(\mathbf{a}_h, \mathbf{a}_{t_k})) + \sum_0^{|H_2|} \varphi(-\cos(\mathbf{a}_h, \mathbf{a}_{t_z})) \right), \quad (2)$$

where $|H| = |H_1| + |H_2|$ is the total number of nodes selected, $|H_1|$ denotes the number of nodes with links, $|H_2|$ denotes the quantity of nodes without links. \mathbf{a}_{t_k} denotes the nodes that are linked to \mathbf{a}_h and \mathbf{a}_{t_z} denotes the nodes that are not linked to \mathbf{a}_h . \cos is a cosine similarity function that measures the similarity between attribute embeddings, assigning larger values to more similar embeddings. ω is a distance-based distinction function that assigns different weights to links based on the number of hops:

$$\omega(h, t_k) = e^{\frac{\lambda}{dis(h, t_k)}}, \quad (3)$$

where λ is a hyperparameter and dis represents the direct hop count between h and t_k . We use φ to compensate for the lack of regularization in the attribute embedding loss:

$$\varphi(x) = \frac{1}{\gamma} \log(1 + e^{-\gamma x + b}), \quad (4)$$

where γ and b are loss margin parameters that can adjust the magnitude of regularization.

3.2.2 Neighborhood Aggregation Stage

Long Distance Interaction Capture Module In the neighborhood aggregation stage, to capture the complete interactions of nodes from complex neighborhoods and expand our model’s cognitive scope, we introduce the R-GCN (Schlichtkrull et al., 2018) module for multi-hop information propagation. The R-GCN module separately assign weight matrices for relationship types and tail node types to enhance the flexibility of the model’s aggregation process. Additionally, to reduce the cognitive noise in the R-GCN layer’s aggregation of node and attribute embeddings, we use two separate sets of R-GCNs to collect information from node embeddings and attribute embeddings, respectively:

$$\mathbf{A}_{h,\tau}^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} \mathbf{a}_{t_j}^{(l)} + W_t^{(l)} \mathbf{A}_{h,\tau}^{(l)} \right), \quad (5)$$

$$\mathbf{S}_{h,\tau}^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c'_{i,r}} W_r'^{(l)} \mathbf{e}_{t_j}^{(l)} + W_t'^{(l)} \mathbf{S}_{h,\tau}^{(l)} \right), \quad (6)$$

where $\mathbf{A}_{h,\tau}^{(l)} \in \mathbb{R}^d$ and $\mathbf{A}_{h,\tau}^{(l+1)} \in \mathbb{R}^d$ are the results of l -layer and $l+1$ -layer attribute aggregation respectively at time τ and the input is attribute embedding $\mathbf{a}_{t_j}^{(l)}$. Correspondingly, $\mathbf{S}_{h,\tau}^{(l)}$ and $\mathbf{S}_{h,\tau}^{(l+1)}$ represent the results of node aggregation used for link prediction, with the input being node embedding $\mathbf{e}_{t_j}^{(l)}$. $c_{i,r}$ is a normalization constant and N_i^r denotes the neighbor set of node i with $r \in \mathcal{R}$. $W_r^{(l)}$ and $W_r'^{(l)}$ represent edge-specific weight matrices. $W_t^{(l)}$ and $W_t'^{(l)}$ indicate the types of special

relationship self-loop matrices for each node.

LLM Aggregation Module To achieve GNN-like aggregation based on LLMs, we design prompts that describe node neighborhood information in the form of links, allowing the LLM to perform importance ranking. This approach effectively leverages the external knowledge and contextual learning capabilities of the LLM. The prompt template is shown in Figure 4.

Our prompt template consists of five parts. First, it provides a description of the temporal evolution attribute graph, specifying the format of neighborhood inputs and clarifying the meaning of variables at each position. The second part instructs the LLM on the task it needs to solve, namely, to prioritize and sort the link identifiers across all neighborhoods and return the results in a list format. The third part includes the identifiers of the nodes to be aggregated, along with all their links at the current timestamp. Next, we inform the LLM of the factors to consider when sorting the links, such as the differences in attributes. Finally, the format constraints for the result are provided, specifying the contents to be included in the returned list.

After obtaining the link bias list, we concatenate the node embeddings, attribute embeddings, and corresponding relation embeddings for all tail nodes in the neighborhood, reorder them according to the list indices, and then compute the weighted average to obtain the aggregated embedding:

$$\mathbf{L}_{h,\tau} = \frac{\sum_{i=1}^N (N-i) [\mathbf{a}_i^\tau; \mathbf{e}_i^\tau; \mathbf{r}_i^\tau]}{\sum_{i=1}^N (N-i)}. \quad (7)$$

Here, \mathbf{a}_i^τ and \mathbf{e}_i^τ represent the attribute embedding and node embedding of the i -th tail node, respectively, while \mathbf{r}_i^τ denotes the relation embedding corresponding to the link. N represents the number of neighbors, where higher-ranked neighbors receive greater weights. It is important to note that for each node to be predicted, the link rankings for all known temporal attribute graphs must be obtained through prompt templates. Finally, the aggregated embeddings are combined into an aggregated sequence for decoding by subsequent modules.

3.2.3 The Prediction Stage

After obtaining the results from the two aggregation modules above, we concatenate them to form the aggregated input sequence for each target node, which is then fed into the GRU to capture temporal

dependencies and generate the predicted embedding:

$$\mathbf{T}_{\mathbf{A}_{h,\tau}} = GRU_{att}([\mathbf{A}_{h,\tau}; \mathbf{L}_{h,\tau}; \mathbf{e}_h], \mathbf{T}_{\mathbf{A}_{h,\tau-1}}), \quad (8)$$

$$\mathbf{T}_{\mathbf{S}_{h,\tau}} = GRU_{node}([\mathbf{S}_{h,\tau}; \mathbf{e}_h; \mathbf{r}_{h,t}], \mathbf{T}_{\mathbf{S}_{h,\tau-1}}). \quad (9)$$

Here, $\mathbf{T}_{\mathbf{A}_{h,\tau}} \in \mathbb{R}^{3d}$ is the attribute prediction embedding at timestamp τ . To predict attributes over multiple time steps, we need to predict the topological information of the attribute graph for future timestamps as auxiliary input. Therefore, $\mathbf{T}_{\mathbf{S}_{h,\tau}} \in \mathbb{R}^{3d}$ represents the node embedding information used for topological prediction. $\mathbf{r}_{h,t}$ represents the relationship embedding associated with the link between the head node h and the tail node t .

Finally, we concatenate the predicted attribute embedding at the future timestamp $\tau + 1$ with the node embedding of the head node h , and decode it through a fully connected layer to obtain the predicted attribute values:

$$\mathbf{a}_h^{\tau+1} = W_{att}[\mathbf{e}_h; \mathbf{T}_{\mathbf{A}_{h,\tau+1}}] + b_{att}. \quad (10)$$

Here, \mathbf{W}_{att} is a learnable weight matrix, and b_{att} is the bias matrix.

For the prediction of graph structures, we input the node embeddings, relationship embeddings, and time-dependent representations of the points between the nodes to be predicted into the fully connected layer. We select the entity with the highest matching degree from the set as the most likely candidate to form a connection:

$$t_{\tau+1} = W_{struct}[\mathbf{e}_h; \mathbf{r}_{h,t}; \mathbf{T}_{\mathbf{S}_{h,\tau+1}}] + b_{struct}. \quad (11)$$

Here, \mathbf{W}_{struct} is the learnable weight matrix for the fully connected layer, and b_{struct} is the bias matrix.

3.2.4 Parameter Learning

We set three loss functions for our LGA framework:

$$\mathcal{L} = \mathcal{L}_a + \lambda \mathcal{L}_{att} + \mu \mathcal{L}_{struct}. \quad (12)$$

Where \mathcal{L}_a represents the attribute embedding loss as indicated in Eq (2), \mathcal{L}_{att} is the MSE loss that measures the attribute accuracy, and \mathcal{L}_{struct} is the cross-entropy loss function that measures the correctness of candidate node prediction. λ and μ are hyperparameters that control the weights of losses.

$$\mathcal{L}_{att} = \frac{1}{N} \sum_{i=1}^N (a_{h_i}^\tau - a_{h_i}'^\tau)^2, \quad (13)$$

$$\mathcal{L}_{struc} = \sum_{i=1}^N \sum_{j=1}^N y_{h_i, t_j} \log(p_{h_i, t_j}). \quad (14)$$

Here, $a_{h_i}^\tau$ is the predicted value of the head node and $a_{h_i}'^\tau$ is the ground truth. y_{h_i, t_j} is the true label and p_{h_i, t_j} is the predicted probability of link existence.

4 Experiments

We evaluated the performance of our LGA framework on the temporal evolution attribute graph prediction task using five real-world open-source datasets from different fields. The LLM we have chosen is Qwen2-7b, which has been pre-trained on graph data. All experiments in this study were conducted on a Tesla A6000 GPU with 48GB of VRAM.

Dataset	#Train	#Valid	#Test	#Nodes	#Rel	#Attribute
CAC	2070	388	508	90	1	1
MTG	270,362	39,654	74,730	44	90	2
NBA	145,876	76,200	42,702	3,859	1	35
ATG	463,188	57,898	57,900	58	178	1
AGG	3,879,878	554,268	1,108,538	6,635	246	1

Table 1: Indicator statistics for the five utilized datasets

4.1 Experimental Setup

4.1.1 Datasets

Considering the influence of graph complexity, number of node attributes and types of relationships, we select five datasets from the real world to verify the predictive ability of the model in different scenarios: CAC¹, MTG², NBA³, ATG⁴ and AGG. The detailed description of the dataset is shown in Appendix A.1 and the specific data of each dataset is shown in Table 1.

4.1.2 Baseline Methods

We compare LGA with two types of models: one that utilizes non-graph-structured assistance and another that uses graph-structured assistance. Among the non-graph structure-assisted models, we choose the ARIMA model, the vector autoregressive model (VAR) and historical average model, which are based on mathematical models; and the Seq2Seq model, which is predicted using a GRU. Among the graph structure-assisted models, we use the HyTE (Dasgupta et al., 2018), TA-DistMult (García-Durán et al., 2018), ConvE (Shi

and Zhao, 2022), RE-NET (Jin et al., 2020), the DARTNET model (Garg et al., 2020) and FC-STGNN (Wang et al., 2024).

4.1.3 Evaluation Metrics

Temporal evolution attribute graph prediction focuses on predicting the attribute values of nodes at future timestamps. To evaluate the accuracy, we use MSE loss to measure the error between the predicted node attributes and the actual attributes. Throughout the training phase of our LGA model, we carefully adjusted the loss proportions in the three components and fine-tuned the dropout parameter to enhance the model’s generalization performance. Finally, we compared our model with other models using the optimal configuration.

4.2 Main Results

In Table 2, we present the performance of our model across five datasets. Our findings highlight the superior performance of our model compared to those that do not incorporate graph structure as auxiliary information, demonstrating that the interactions between nodes in the topology influence their own attributes. Furthermore, when compared to models that treat graph structure as auxiliary information, our model achieves higher attribute prediction accuracy, indicating that it better captures the intrinsic semantics of attributes and reduces noise in neighborhood extraction. In contrast, simple node embedding and attribute embedding predictions based on concatenation struggle to capture more effective information. Lastly, a comparison with FC-STGNN shows that our model also possesses the capability to understand the interactions between attributes.

4.3 Study on the Relationship Between Attributes and Topology

To verify the relationship between node attribute differences and inter-node hop distances, we conducted experiments using the CAC and MTG datasets with three hops. We systematically enumerated all node pairs within one-hop, two-hop, and three-hop neighborhoods in the graph and quantified their attribute differences using Euclidean distance measurements. To accurately capture the changes in MTG, we utilize the last three digits of attribute differentials for visualization purposes. The experimental results which are shown in Figure 3 indicate that the difference in node attributes exhibits positive (or negative) linear relationship

¹<https://www.aminer.cn/>

²<https://www.imf.org>

³<https://www.basketball-reference.com>

⁴<https://www.imf.org>

	Method	ATG (10^{-3})	CAC (10^{-2})	MTG (10^{-4})	AGG (10^{-4})	NBA (10^{-2})
w/o graph	Historic Average	1.636	4.540	14.930	600.000	-
	VAR	3.961	6.423	9.490	300.000	-
	ARIMA	1.463	4.245	2.860	51.240	-
	Seq2Seq model	1.323	4.554	2.975	28.000	-
with graph	HyTE + 1-layer GRU	4.041	40.234	37.170	7.430	-
	HyTE + 2-layer GRU	1.531	40.885	17.410	2.070	-
	TA-Distmult + 1-layer GRU	0.847	3.584	16.880	3.250	-
	TA-Distmult + 2-layer GRU	0.796	3.432	9.770	7.030	-
	RENet (mean) + 1-layer GRU	0.793	4.073	5.020	203.320	7.870
	RENet (mean) + 2-layer GRU	0.857	3.865	4.348	200.220	-
	RENet (RGCN) + 1-layer GRU	0.620	3.718	5.170	203.120	<u>5.425</u>
	RENet (RGCN) + 2-layer GRU	0.550	3.984	12.700	201.560	-
	ConvE + 1-layer GRU (Shi and Zhao, 2022)	0.763	3.899	7.240	202.580	-
	ConvE + 2-layer GRU (Shi and Zhao, 2022)	0.728	4.321	9.460	206.640	-
	DARTNET (Garg et al., 2020)	0.115	3.423	0.496	0.848	9.205
	FC-STGNN (Wang et al., 2024)	0.892	5.565	0.958	1.324	8.937
	LGA(ours)	0.079	1.934	0.480	0.160	5.048

Table 2: Temporal evolution attribute graph prediction results of all models. Bold formatting indicates the best results, and underlined formatting denotes the second-best results.

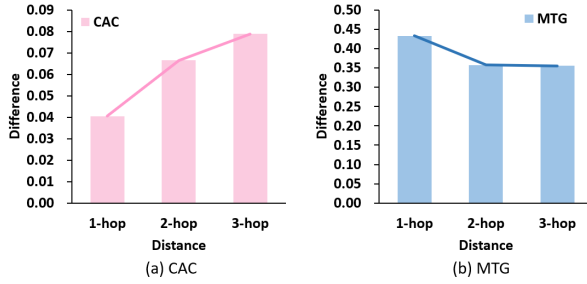


Figure 3: The connection between node attribute differences and inter-node hop distances

with inter-node hop distances. Therefore, incorporating the topology of nodes as semantic information into the attribute embeddings can enhance the representational power of the attributes.

4.4 Study on the GNN-based Prediction Performance of LLM

In this experiment, we examine the aggregation performance of LLMs in temporally evolving attributed graphs and their compatibility with various models by integrating a GNN-style aggregation module from LLMs into baseline approaches that partially leverage graph structures as auxiliary information. The experimental results are shown in Table 3.

Experimental results demonstrate that the LLM-based aggregation module can more accurately capture relationships between adjacent nodes, comprehend the temporal evolution of node attributes, and consequently enhance the prediction accuracy of

node attributes.

Dataset	CAC (10^{-2})	NBA (10^{-2})	ATG (10^{-3})	AGG (10^{-4})	MTG (10^{-4})
DARTNET	3.423	9.205	0.115	0.848	0.496
DARTNET + LLM	2.443	6.837	0.103	0.530	0.190
RE-NET(mean)	3.865	7.870	0.857	200.22	4.348
RE-NET(mean) + LLM	3.485	7.750	0.305	0.740	1.970
RE-NET(RGCN)	3.718	5.425	0.620	203.12	5.170
RE-NET(RGCN) + LLM	2.647	5.098	0.134	0.161	1.860
FC-STGNN	5.811	8.937	0.111	0.912	0.998
FC-STGNN + LLM	5.653	8.546	0.226	0.811	0.951

Table 3: Study on the GNN-based Prediction Performance of LLM

4.5 Time Complexity Analysis

The main time cost of LGA lies in the usage of R-GCN and LLM. Therefore, we conducted the following time complexity analysis based on these two components. We first define the following parameters for our dataset and model architecture: N : Number of nodes in the dataset; E_{node} : Average number of neighbors per node; n : Historical time length; L_{RGCN} : Number of layers in the R-GCN; d_{LLM} : Hidden dimension of the LLM; L_{LLM} : Number of layers in the LLM. For each node attribute prediction step, the time complexity of the R-GCN aggregation is $O(n \cdot L_{RGCN} \cdot N)$. Since the LLM does not participate directly in training, we perform LLM-based aggregation during the data preprocessing stage and cache the results for training. Consequently, the time complexity of LLM aggregation is $O(L_{LLM} \cdot d_{LLM} \cdot N \cdot E_{node}^2)$. Thus,

the overall time complexity is: $O(n \cdot L_{RGCN} \cdot N + L_{LLM} \cdot d_{LLM} \cdot N \cdot E_{node}^2)$. Given that n , L_{RGCN} , L_{LLM} , and d_{LLM} are predefined constants, our model’s time complexity scales primarily with the graph size N and its sparsity E_{node} .

At the same time, we calculated the time proportion of the R-GCN and LLM processing processes in a single epoch through experiments, as shown in Table 4. Among the four evaluated datasets—AGG, NBA, CAC, and MTG—the sparsity decreases in the order of AGG (sparsest), NBA, CAC, and MTG (densest), while the scale increases from MTG (smallest) to CAC, AGG, and NBA (largest). Experimental results demonstrate that (1) larger attribute graphs allocate a greater time proportion to R-GCN processing, and (2) sparser graphs exhibit higher LLM aggregation overhead, though absolute runtime remains significant. Nevertheless, the proposed method effectively balances computational demands, maintaining manageable complexity despite jointly employing LLM and R-GCN.

Dataset	total (/s)	w/o LLM (/s)	w/o RGCN (/s)
CAC	249	40(-83.9%)	215(-13.7%)
AGG	3649	124(-96.6%)	3616(-13.7%)
NBA	39627	5386(-86.4%)	39149(-1.2%)
MTG	38399	21811(-43.2%)	32013(-16.6%)

Table 4: Time Complexity Analysis

Dataset	CAC (10^{-2})	NBA (10^{-2})	ATG (10^{-3})	AGG (10^{-4})	MTG (10^{-4})
W/o AttLoss	2.186	7.770	0.090	12.628	0.510
W/o LLM	2.731	8.468	0.095	0.540	1.822
W/o Long-distance Capture	2.132	8.032	0.112	0.840	1.370
LGA	1.934	5.048	0.079	0.160	0.480

Table 5: Performance of ablation study

4.6 Ablation Study

To substantiate the effectiveness of each module, we conducted ablation experiments, systematically breaking down and analyzing our proposed solution. As shown in Table 5, the optimal result is the sum of all the components. From one perspective, the most significant decline in prediction accuracy observed after removing the LLM module underscores its crucial role as an information aggregation method in the model’s prediction process. From another perspective, the structural semantic enhancement of attributes and the long-range interaction capture module provide additional insights from

both structural and temporal dimensions, playing a secondary yet indispensable role.

5 Conclusion

This paper presents LGA, a novel model for temporal evolution attribute graph prediction that enhances accuracy and stability. Specifically, we enrich the semantic meaning of the initial attribute embeddings by constructing an attribute embedding loss, providing the model with topological semantics as prior knowledge. By incorporating the long-distance interactive capture module, we account for multi-hop interactions in complex structures, while separately considering the differences between node and attribute embeddings through two distinct sets of modules. Finally, we design prompts to perform GNN-style aggregation based on LLMs, filling a gap in the field. Comprehensive experiments across multiple datasets show LGA significantly outperforms baseline models.

6 Limitations

Our model utilizes a GNN module for information aggregation, which causes the training time and GPU memory usage to increase as the graph size grows. And the embedding stage’s training time accelerates dramatically with graph complexity due to the combinatorial explosion of node interactions. Additionally, as the historical time length increases, the model requires more GPU memory. In summary, our next research direction is to adapt LGA for larger-scale attribute graphs while minimizing computational resource consumption.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2023YFF0905503, National Natural Science Foundation of China under Grants No.62472188.

References

- Davide Andreoletti, Sebastian Troia, Francesco Musumeci, Silvia Giordano, Guido Maier, and Massimo Tornatore. 2019. [Network traffic prediction based on diffusion convolutional recurrent neural networks](#). In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2019, Paris, France, April 29 - May 2, 2019*, pages 246–251. IEEE.
- Kamal Berahmand, Elahe Nasiri, Mehrdad Rostami, and Saman Forouzandeh. 2021. [A modified deep-](#)

- walk method for link prediction in attributed social network. *Computing*, 103(10):2227–2249.
- Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. 2023. [LLM4TS: two-stage fine-tuning for time-series forecasting with pre-trained llms](#). *CoRR*, abs/2308.08469.
- Junyoung Chung, Çağlar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. [Empirical evaluation of gated recurrent neural networks on sequence modeling](#). *CoRR*, abs/1412.3555.
- Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha P. Talukdar. 2018. [Hyte: Hyperplane-based temporally aware knowledge graph embedding](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2001–2011. Association for Computational Linguistics.
- Valentin Flunkert, David Salinas, and Jan Gasthaus. 2017. [Deepar: Probabilistic forecasting with autoregressive recurrent networks](#). *CoRR*, abs/1704.04110.
- Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. [Learning sequence encoders for temporal knowledge graph completion](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4816–4821. Association for Computational Linguistics.
- Sankalp Garg, Navodita Sharma, Woojeong Jin, and Xiang Ren. 2020. [Temporal attribute prediction via joint modeling of multi-relational structure evolution](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2785–2791. ijcai.org.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021. [Explainable subgraph reasoning for forecasting on temporal knowledge graphs](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020. [Graph hawkes neural network for forecasting on temporal knowledge graphs](#). In *Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020*.
- Yu Hao, Xin Cao, Yixiang Fang, Xike Xie, and Sibow Wang. 2020. [Inductive link prediction for nodes having only attribute information](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1209–1215. ijcai.org.
- Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2023. [Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning](#). In *The Twelfth International Conference on Learning Representations*.
- Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024. [Can GNN be good adapter for llms?](#) In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 893–904. ACM.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. [Recurrent event network: Autoregressive structure inference over temporal knowledge graphs](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6669–6683. Association for Computational Linguistics.
- Yong-Chao Jin, Qian Cao, Ke-Nan Wang, Yuan Zhou, Yan-Peng Cao, and Xi-Yin Wang. 2023. [Prediction of COVID-19 data using improved ARIMA-LSTM hybrid forecast models](#). *IEEE Access*, 11:67956–67967.
- Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xing Xie. 2011. [Discovering spatio-temporal causal interactions in traffic data streams](#). In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1010–1018. ACM.
- Yunpu Ma, Volker Tresp, and Erik A. Daxberger. 2019. [Embedding models for episodic knowledge graphs](#). *J. Web Semant.*, 59.
- Elahe Nasiri, Kamal Berahmand, and Yuefeng Li. 2023. [Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks](#). *Multim. Tools Appl.*, 82(3):3745–3768.
- Athanasios Salamanis, Polykarpos Meladianos, Dionysios D. Kehagias, and Dimitrios Tzovaras. 2015. [Evaluating the effect of time series segmentation on starima-based traffic prediction model](#). In *IEEE 18th International Conference on Intelligent Transportation Systems, ITSC 2015, Gran Canaria, Spain, September 15-18, 2015*, pages 2225–2230. IEEE.
- Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. [Modeling relational data with graph convolutional networks](#). In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.
- Ming Shi and Jing Zhao. 2022. [A knowledge graph link prediction model with combined 1d and 2d convolutional embeddings](#). In *9th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2022, Shenzhen, China, October 13-16, 2022*, pages 1–2. IEEE.
- Volker Tresp, Yunpu Ma, Stephan Baier, and Yinchong Yang. 2017. [Embedding learning for declarative memories](#). In *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I*, volume 10249 of *Lecture Notes in Computer Science*, pages 202–216.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. [Can language models solve graph problems in natural language?](#) In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yucheng Wang, Yuecong Xu, Jianfei Yang, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. 2024. [Fully-connected spatial-temporal graph for multivariate time-series data](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 15715–15724. AAAI Press.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. 2021. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34:28798–28810.

Yuxin You, Zhen Liu, Xiangchao Wen, Yongtao Zhang, and Wei Ai. 2024. [Large language models meet graph neural networks: A perspective of graph mining](#). *CoRR*, abs/2412.19211.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, Simin Wu, and Wenwu Zhu. 2023. [Llm4dyg: Can large language models solve problems on dynamic graphs?](#) *CoRR*, abs/2310.17110.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2022. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*.

A Supplementary Experimental Setup

A.1 Datasets

In this section, we will provide a detailed description of the meanings of the nodes, edges, and attributes of the datasets we are using:

CAC Coauthorship-Citation(CAC): CAC represents the citation information of papers between authors. Among them, nodes represent authors, edges represent collaborative relationships between authors, and attributes represent the number of times authors are cited each year.

MTG Multiattributed Trade Graph(MTG): MTG represents the international trade situation. Among them, nodes represent countries, edges represent export trade orders, and attributes represent monthly net export price index and international reserve asset value.

NBA NBA represents the team formation status between players. Among them, nodes represent different players, edges represent team relationships between players, and attributes represent information such as player scores, teams, ages, and positions.

AGG Attributed GDELT Graph (AGG): AGG represents Global Database of Events, Language, and Tone (GDELT). Among them, nodes represent countries, edges represent events, and attributes represent the currency exchange rate.

ATG Attributed Trade Graph (ATG): ATG represents the export trade between countries. Among them, nodes represent countries, edges represent export trade orders, and attributes represent the currency exchange rate.

B Supplementary Experiments

B.1 Study on Long Distance Interactive Capture

In this experiment, we conducted a comparative analysis using the inclusion of a long-range neighborhood interaction capture module as a benchmark, aiming to investigate the impact of long-range interactions and individual modeling of relationship categories on prediction performance. The results are shown in Table 7. The experimental results show that the long distance interactive capture module can improve the completeness of the model’s acquisition of neighborhood knowledge, thereby enhancing prediction accuracy.

B.2 Study on Attribute Graph Link Prediction

In the task of predicting temporal attributes, modeling structural information for attribute prediction can significantly enhance the accuracy of attribute predictions. Therefore, as a secondary focus, we executed experiments to assess the graph structure prediction accuracy against the DARTNET model within the context of the temporal attribute prediction task. This was performed on the CAC, ATG, and MTG datasets to evaluate the model’s

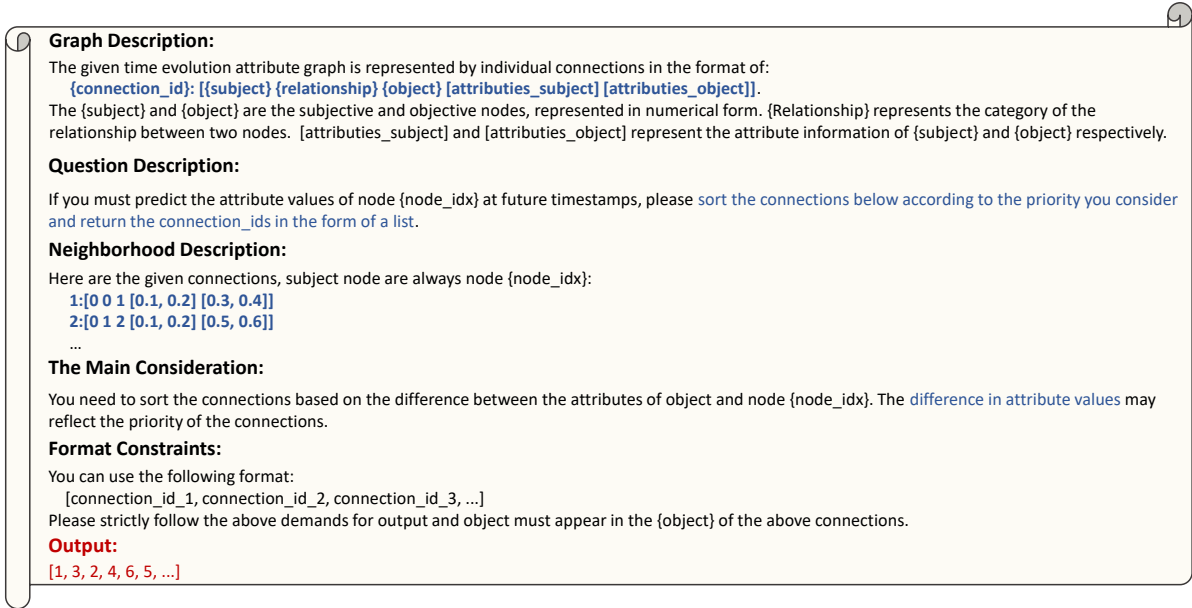


Figure 4: The prompt template collects the semantic information of neighborhood events

Dataset	CAC				ATG				MTG			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
DARTNET	53.64	46.69	56.93	65.36	91.55	88.09	93.87	97.82	85.86	80.26	89.79	95.44
LGA	54.16	47.29	55.42	69.28	92.78	89.86	94.71	98.11	89.35	83.31	94.77	98.98

Table 6: Performance of attribute graph structure prediction study on CAC, ATG and MTG datasets

Dataset	CAC (10^{-4})	ATG (10^{-4})	MTG (10^{-3})
Origin	3.423	0.115	0.496
W.Long-distance Capture	2.443	0.090	0.195

Table 7: Study on long distance interactive capture

capability to generate structural auxiliary information. We use the mean reciprocal rank (MRR) and Hits@1/3/10 as performance metrics for structural prediction. As depicted in Table 6, LGA can accurately predict graph structures and offer more valuable neighborhood information.

B.3 Time Step Study

Our model can predict multiple future time steps through continuous prediction, and to verify the effectiveness of its multistep prediction ability, we predict the attribute prediction errors for 1, 2, and 5 consecutive unknown time steps and use the CAC, ATG, MTG, and NBA for experiments.

Fig. 5 shows that our LGA model yields varying degrees of improvement over the baseline in terms of the stability of its multistamp predictions. On the small-scale CAC dataset, the loss increases

rapidly because the model does not learn complete information, but its increase is smaller than that of the baseline. On the larger ATG and MTG datasets, the increase rate of the prediction loss induced by our model after 2 timestamps significantly slows, and it even exhibits negative growth on the MTG data, which is sufficient to illustrate the multistamp prediction stability advantage of LGA.

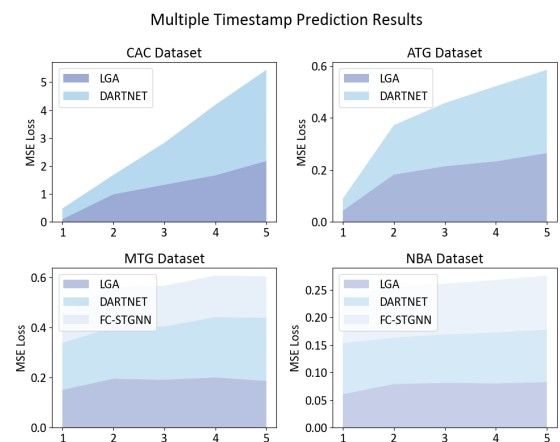


Figure 5: The multiple timestamp prediction results obtained from the tested models

B.4 Prompt Sensitivity Experiment

The problem-solving capability of LLMs is highly dependent on the quality of prompt template design. Creating superior prompt templates can enhance the model’s comprehension ability and stability. Therefore, in this experiment using the CAC dataset, we focused on two components that most significantly impact LLM performance - Question Description and Format Constraints - by modifying their prompt wording to investigate the model’s sensitivity to different prompt formulations.

First, regarding the Question Description component of the prompt template, we modified it to: "Now you should predict the attributes of subject node *node_idx*. By analyzing the provided connections, identify the most relevant relationships to the subject node and rank them by importance (highest priority first)." In attribute prediction experiments, this modification resulted in an MSE loss of 2.216, representing a 14.6% performance degradation compared to the original result of 1.934, indicating lower adaptability of the modified template to the LLM.

Next, we examined the role of Format Constraints in the prompt template. The experiment attempted to remove the constraint phrase "Please strictly follow the above demands for output" that forces the LLM to return results in a specific format. However, during testing, we observed that the LLM returned unstructured intermediate reasoning outputs in some aggregation scenarios. This anomalous behavior directly disrupted subsequent data aggregation processes, demonstrating the irreplaceable role of Format Constraints in regulating LLM output formats.

In conclusion, through experimental validation of different prompt templates, we demonstrate that the prompt template proposed in this study delivers optimal performance.

B.5 Embedding Effectiveness Evaluation

To evaluate the effectiveness of attribute embedding, we conducted a focused analysis on node 79 and its neighborhood (nodes [53, 78, 48]) at timestamp 2003 in the CAC dataset. In the baseline model without attribute embedding, the Euclidean distances between node 79’s feature representation (transformed through the connection layer) and its neighbors were [0.0038, 0.3920, 0.3413]. After implementing attribute embedding, these distances reduced significantly to [0.0015, 0.1526, 0.1345].

This quantitative improvement demonstrates that the attribute embedding loss function effectively enhances feature similarity between connected nodes in the embedding space.