Modeling Interactions Between Stocks Using LLM-Enhanced Graphs for Volume Prediction

Zhiyu Xu[1], Yi Liu[1], Yuchi Wang[1], Ruihan Bao[2], Keiko Harimoto[2], Xu Sun[1]
[1]National Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University
[2]Mizuho Securities Co., Ltd.
zhiyu_xu@stu.pku.edu.cn, imliuyi@pku.edu.cn, wangyuchi@stu.pku.edu.cn
{ruihan.bao, keiko.harimoto}@mizuho-sc.com, xusun@pku.edu.cn

Abstract

Accurate trading volume prediction is essential for portfolio optimization, market regulation, and financial risk control. An effective method for predicting trading volume involves building a graph to model relations between stocks. Recent research has enhanced these models by integrating stock news to improve forecasting ability. However, existing approaches primarily integrate news data as auxiliary features for nodes in Graph Neural Networks (GNNs), often overlooking the relational information between stocks embedded in news. To address this, we propose LLM-Enhanced Dynamic Graph Neural Network (LED-GNN), a framework that constructs dynamic graphs using inter-stock relationships extracted from news via a large language model (LLM)-centered pipeline. The news graph is then combined with graphs learned from historical price-volume data and fed into a dynamic GNN to generate predictions. Evaluated on a real-world dataset, TOPIX, with Reuters Financial News, LED-GNN consistently outperformed all baseline models, achieving a 2% improvement over the strongest baseline.

1 Introduction

Trading volume refers to the total amount of stock transaction within a certain unit of time. The prediction of the trading volume is of significant value in portfolio optimization, marketing regulation, and financial risk control (Brownlees et al., 2010). Historically, many significant market changes have been accompanied by unusually high trading volumes, such as "Black Monday" in 1987 (Shiller, 1987; Gallant et al., 2015). Trading volume prediction can be beneficial for developing stock trading strategies, as substantial orders can push the stock price in an unfavorable direction for the investor (Białkowski et al., 2008). This shift in stock price can be mitigated by dividing large positions according to accurate knowledge of future volume trends, thus achieving higher investment profits.

Graph neural networks (GNNs) have garnered increasing attention in stock prediction for their ability to model inter-stock relations (Sawhney et al., 2021b; Kim et al., 2019). Since stock data lacks inherent graph structures, various methods are employed to construct graphs, including utilizing prior knowledge (Kim et al., 2019; Zheng et al., 2023) and mining relational data from historical stock prices and trading volumes (Xiang et al., 2022; Sawhney et al., 2021a; Li et al., 2022).

Incorporating external information (Lo, 2004), such as news data, has also shown great potential in improving prediction accuracy. Sometimes, news data are integrated with graph neural networks as auxiliary features for node representation (Zhao et al., 2021). However, managing redundancy and noise in lengthy news articles remains a persistent challenge, prompting the development of various methods to extract key information (Liang et al., 2020; Zou et al., 2022; Zhou et al., 2021). Recently, with the emergence of large language models (LLMs), financial expert LLMs have been developed to enhance understanding of news data and provide more informed investment advice (Liu et al., 2023).

However, a crucial point remain unexplored. A significant portion of news pertains to multiple stocks and the relationships among them, naturally forming an implicit graph with stocks as nodes. While considerable efforts have been made to model stock data using graphs, news data are typically incorporated only as a part of the node features. By modeling news as graph edges, it becomes possible to capture the impact of sudden events on the relations between stocks. However, the potential of leveraging relational news information directly in the graph's edges remains largely unexplored. Yet, extracting meaningful relationships from lengthy and complex news articles

Proceedings of the Joint Workshop of the 9th FinNLP, the 6th FNP, and the 1st LLMFinLegal, pages 153–163
January 19–20, 2025. ©2025 Association for Computational Linguistics

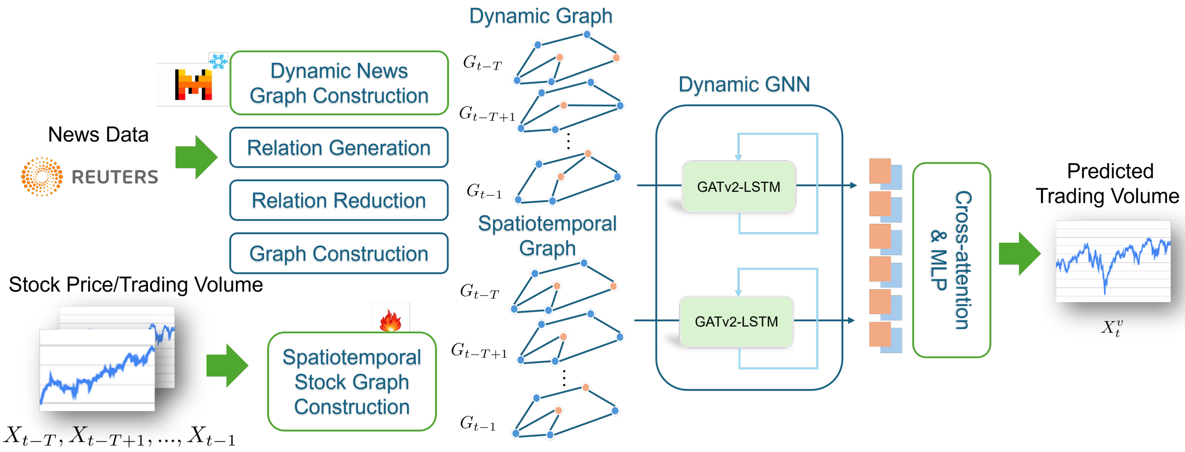


Figure 1: An overview of the proposed LED-GNN framework begins with constructing a dynamic news graph from news data through three phases: relation generation, relation reduction, and dynamic graph construction. Additionally, a spatiotemporal stock graph is learned from historical price and trading volume data. These two graphs are then processed by a dynamic GNN, where node representations are integrated using cross-attention, and final predictions are produced via a multi-layer perceptron (MLP).

poses significant challenges, particularly due to the scarcity of specific and labeled training data. To address this, we propose a Large Language Model (LLM)-enhanced pipeline tailored to effectively extract relational information. Building on this, we introduce LLM-Enhanced Dynamic Graph Neural Network (LED-GNN), a framework for predicting trading volume more effectively. Our approach constructs a dynamic graph using relationships between stocks derived from news articles via the LLM-enhanced pipeline. Additionally, a graph structure is learned from stock price and volume data. Both graphs are then processed through a generic GNN architecture designed for dynamic graphs, producing node embeddings that are subsequently utilized for predictions. The specific framework of our method is shown in Figure 1.

We evaluated our model on a real-world dataset, TOPIX (Zhao et al., 2021), along with news data collected from Reuters Financial News. Our model consistently outperformed all baselines, achieving a 2% improvement over the strongest baseline model.

In summary, our contributions are as follows:

- We propose a pipeline for constructing dynamic news relation graphs with large language models, leveraging their exceptional understanding of natural language and the domain knowledge acquired during pre-training to process stock news articles. To the best of our knowledge, this is the first work to use relationships extracted from stock news as edge features to construct a dynamic graph for stock volume prediction.

- To coordinate the relationships learned from historical price-volume data and news data, we propose LED-GNN, which is capable of handling the dynamic news graph and spatiotemporal stock graph and aligning data from both sources for accurate predictions.

- We conducted volume prediction experiments using a real-world dataset comprising TOPIX stock price-volume data and related news. In these experiments, LED-GNN consistently demonstrated superior performance compared to all baseline models.

2 Methdology

2.1 Problem Definition

The problem of trading volume prediction can be viewed as a regression problem. Consider a multivariate time series $X = \{X_{:,1}, X_{:,2}, ..., X_{:,t}, ...\}$, where each time slice $X_{:,t} = \{x_{1,t}, x_{2,t}, ..., x_{S,t}\} \in R^{S\times C}$ represents the state of all S stocks at time t. The C-dimensional feature vector $x_{i,t} \in R^{1\times C}$ for each stock i describes the overall characteristics of a single stock at a given time, consisting of stock volume data $x^v_{i,t}$ and price data $x^p_{i,t}$ including the highest price, lowest price, opening price, and closing price.

Given a time window $W = \{t, t+1, ..., t+T_0\}$ of length T_0, with known news data $\mathcal{D}_{t:t+T_0}$

and time series data $\mathcal{X}_{t:t+T_0}$ for all S stocks, our objective is to predict the trading volume $X^v_{:,t+T_0+1}$ for each stock in the next time step (in our case, one hour later).

To accurately capture the relationships between stocks, the data is further modeled as a dynamic graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, where the node set $\mathcal{V}$, which represents all S stocks remain static, and the edge set $\mathcal{E}_t$, which is derived from historical news, stock prices and trading volumes, evolves over time. Thus, the problem is formalized as finding the function $\mathcal{F}_\theta$ such that:

$$X^v_{:,t+T_0+1} = \mathcal{F}_\theta(\mathcal{G}_{t:t+T_0}, \mathcal{D}_{t:t+T_0}, \mathcal{X}_{t:t+T_0}) \quad (1)$$

Here, the edge set $\mathcal{E}_t$ of the graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ is a function of the historical data $\mathcal{X}_{:t+T_0}$ and news data $\mathcal{D}_t$, i.e., $\mathcal{E}_t = \phi_{\theta_1}(\mathcal{X}_{:t+T_0}, \mathcal{D}_t)$.

2.2 Model Overview

As shown in Figure 1, LED-GNN consists of three main components. The news relation graph module processes news data, constructing the graph through three phases: relation generation, relation reduction, and graph construction. The stock graph construction module builds a spatio-temporal graph from stock price and trading volume data. These two graphs are then input into a dynamic GNN and GATv2-LSTM is incorporated to learn node representations and a cross-attention layer to align the representations from both graphs.

2.3 News Relation Graph Construction

The news data for the stocks consist of the news title, the entities (the stocks or companies mentioned in the news), and the news body. News articles are typically document-level corpora, averaging over 300 words, and tend to be sparse in terms of the relationships they imply. Furthermore, unlike relation extraction datasets such as DocRE (Yao et al., 2019), the absence of a predefined relation set poses another challenge. To address these problems and extract the underlying relations between stocks, we designed the following pipeline as shown in Figure 2. The pipeline includes three steps: relation generation, relation reduction, and dynamic graph construction. It should be noted that, taking both accuracy and efficiency into account, Mistral-7B (Jiang et al., 2023) is used as the backbone and its parameters remain frozen.

2.3.1 Relation Generation Phase

During the relation generation phase, part of the news dataset is selected and inputted into the large language model. In the news article, the relations between each two entities are extracted, generating a set of relations R'.

To better utilize large language model's ability to interpret the news article, we design a one-shot prompt composed of instructions, a given example of relation generation, the entities mentioned in the news and the news article. (Figure 3)

2.3.2 Relation Reduction Phase

The relation set R' generated by the relation generation phase may contain redundant expressions. The relation reduction phase address this problem by adopting a framework proposed by Grootendorst (2022), obtaining a more concise relation set R.

As shown in Figure 2, during the relation reduction phase, Sentence-BERT (Reimers and Gurevych, 2019) converts relational phrases in R' into dense, high-dimensional vector representations. The Uniform Manifold Approximation and Projection (UMAP) method (McInnes et al., 2020) reduces the dimensionality of these embeddings while preserving global and local features (Grootendorst, 2022; McInnes et al., 2020; Allaoui et al., 2020). The HDBSCAN algorithm (McInnes and Healy, 2017) then performs soft clustering by automatically determining the number of clusters for semantically similar relations and filtering out unrepresentative categories, resulting in a more concise set of relations R.

2.3.3 Dynamic Graph Construction Phase

The dynamic graph construction module leverages an LLM to generate relation triplets from news in the format (subject, relation, object). These triplets are then converted into edge feature vectors using one-hot encoding, creating a dynamic relation graph from stock news.

Triplet Generation The triplet generation phase uses the same prompt structure as in the relation generation phase (Section 2.3.1), but requires the LLM to select a relation from the predefined relation set R (shown in purple in Figure 3).

Each triplet extracted from the news is denoted as $< E_i, r_k, E_j >$, where E_i and E_j are entities representing companies or stocks, and r_k is a relation from R that the LLM selects to describe the relationship between E_i and E_j. Due to the inherent randomness and potential hallucinations of

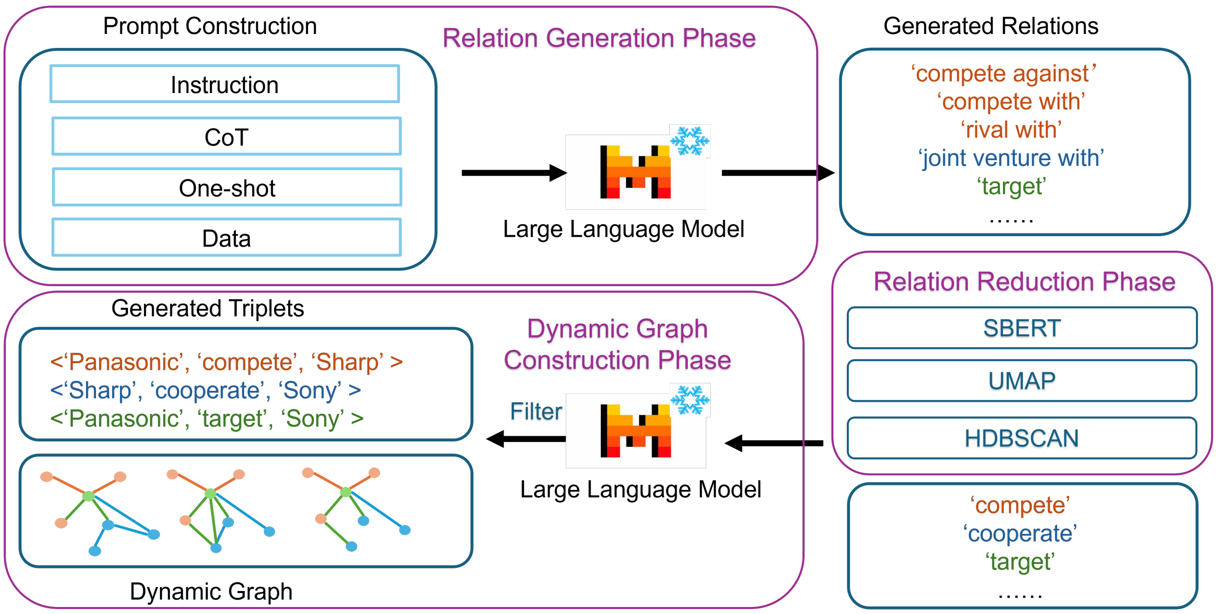


Figure 2: An illustration of the pipeline for constructing the news relation graph.

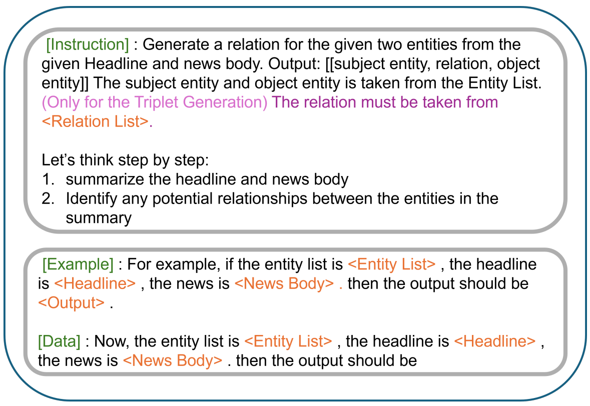


Figure 3: This illustration depicts the structure of the prompt. The Relation List (shown in purple) is included only during the Triplet Generation and is not yet added during the Relation Generation Phase.

large language models (Huang et al., 2023), some of the generated relations may not be present in R. To address this, a filtering process matches the generated relation r_k with similar relations in R. If no match is found, r_k is added to R for future generation. Notably, the relation set R includes a "no relation" option to handle cases where the news does not describe a relationship between the entities.

Dynamic Graph Construction After the relation triplets $T < E_i, r_k, E_j >$ for each news are generated, one-hot encoding is used to map the relation r_k to a $card(R)$-dimensional vector $\mathbf{u}$. For a given time t_0 and a lookback window of size T, the edge feature between entities E_i and E_j is computed with the relation triplets from news that occurred during this time period as follows:

$$e_{t_0,ij}^{News} = \sum_{t=t_0-T_1}^{t_0} \sum_k u_{k,t} \tag{2}$$

where $\sum_k u_{k,t}$ denotes the summing of the every one-hot vector derived from the news at the time t. Since the impact of news on stocks is typically short-term (within a day), T is set to 24 hours. This process constructs a dynamic news relation graph, denoted as $\mathcal{G}_t^{\text{News}} = (\mathcal{V}, \mathcal{E}_t^{\text{News}})$.

2.4 Spatiotemporal Stock Graph Construction

The spatiotemporal graph construction module learns the graph distribution from historical data on stock prices and trading volumes in an end-to-end fashion. We adopt the graph learning approach proposed by Shang et al. (2021), where the adjacency matrix A is modeled as a random variable drawn from a matrix Bernoulli distribution parameterized by θ, such that $A = \phi(\theta)$. To address the discreteness of A_{ij}, the Gumbel reparameterization trick (Jang et al., 2017) is used to make $\phi(\cdot)$ differentiable. In this framework, θ is derived from the historical feature sequence (not restricted to the lookback window) of the nodes by a link predictor.

It is important to note that the resulting graph structure is a spatiotemporal graph, not the dynamic graph discussed in Section 2.3. Unlike

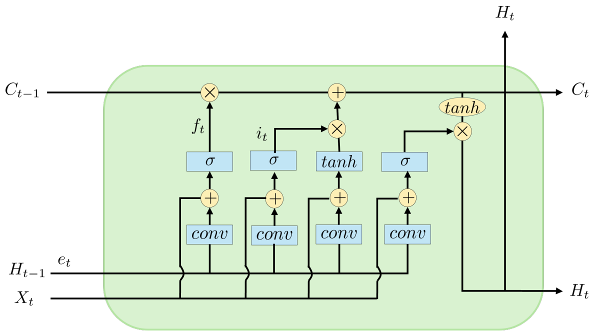


Figure 4: Each cell in GATv2-LSTM.

the news relation graph, the edge structure in the stock spatiotemporal graph does not change and only the node features vary over time. Specifically, the spatiotemporal graph can be denoted as $\mathcal{G}_t^{\text{Stock}} = (\mathcal{V}, \mathcal{E}^{\text{Stock}})$.

2.5 Dynamic Graph Neural Network

In this section, we predict future trading volume trends using a dynamic graph neural network that processes the dynamic news relation graph (Section 2.3) and the spatiotemporal stock graph (Section 2.4). As shown in Figure 1, GATv2-LSTM generates node representations for both graphs, which are then fused through a cross-attention layer and passed to an MLP for final prediction.

To handle graphs where node features and edges evolve over time, we adopt an approach that combines temporal and spatial propagation (Seo et al., 2016). Information is propagated alternately through a graph neural network and a recurrent neural network to generate node representations. As illustrated in Figure 4, each LSTM-like cell updates node states by aggregating information from neighbors and propagating the updated representations along the time dimension. The process is mathematically defined as:

$$I_t = \sigma(X_t \cdot W_i + \text{conv}_i(H_{t-1}, e_t) + b_i) \quad (3)$$
$$F_t = \sigma(X_t \cdot W_f + \text{conv}_f(H_{t-1}, e_t) + b_f) \quad (4)$$
$$\tilde{C}_t = \tanh(X_t \cdot W_c + \text{conv}_c(H_{t-1}, e_t) + b_c) \quad (5)$$
$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \quad (6)$$
$$O_t = \sigma(X_t \cdot W_o + \text{conv}_o(H_{t-1}, e_t) + b_o) \quad (7)$$
$$H_t = O_t \odot \tanh(C_t) \quad (8)$$

Here, $\text{conv}_i, \text{conv}_f, \text{conv}_c, \text{conv}_o$ represents the GNN modules. For the dynamic news relation graph, e_t are the edge features e_t^{News} extracted from the news, while for the spatio-temporal stock graph, e_t are the edges e^{Stock} obtained in Section 2.4.

For the graph neural network (GNN) part, we use GATv2 (Brody et al., 2022) which is an improved version of the GAT (Veličković et al., 2018) architecture. To prevent static attention from hindering the propagation process, GATv2 applies the attention vector after LeakyReLu.

$$\psi(i,j) = \mathbf{a}^\top \text{LeakyReLU}\left(\mathbf{\Theta}_s \mathbf{x}_i + \mathbf{\Theta}_t \mathbf{x}_j + \mathbf{\Theta}_e \mathbf{e}_{i,j}\right) \quad (9)$$

$$\alpha_{i,j} = \frac{\exp\left(\psi(i,j)\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(\psi(i,k)\right)} \quad (10)$$

After the GATv2-LSTM model computes node embeddings for both the dynamic news relation graph and the spatiotemporal stock graph, a cross-attention mechanism is applied to combine these embeddings. The resulting fused representation is then fed into an MLP to generate the final prediction.

2.6 Objective

The loss function is defined as:

$$\text{loss} = \alpha \cdot \text{loss}_{\text{MAE}} + \beta \cdot \text{loss}_{\text{MSE}} + \lambda \cdot \sum_j \theta_j^2 \quad (11)$$

where loss_{MAE} is the Mean Absolute Error, loss_{MSE} is the Mean Squared Error, θ_j represents the model parameters, and α, β, and λ are hyperparameters.

3 Experimentation

3.1 Experimental Settings

3.1.1 Dataset

Dataset Overview We conducted experiments on the Tokyo Stock Exchange (TOPIX500) (Zhao et al., 2021) dataset, which includes stock data for 500 companies from January 4, 2013, at 9:00 AM to October 1, 2018, at 3:00 PM. The dataset comprises historical stock prices (opening, closing, highest, and lowest), trading volumes, and 146,474 pieces of textual news. Each news entry includes a headline, a body, and relevant stock identifiers, with an average news body length of 391 words. The dataset was split along the time axis in a 4:1 ratio for training and testing.

Data Preprocessing We refine the TOPIX500 dataset by removing stocks with insufficient data, resulting in a final selection of 439 stocks. Similar to many financial datasets, the stock price and trading volume data exhibit a long-tail distribution. To mitigates this and ensure that the stock price and trading volume data are on the same scale, we follow previous work (Zheng et al., 2023) and apply log return to normalize the stock price and trading volume. The specific formula is as follows:

$$p_t = \log(\frac{p_t}{p_{t-1}}) \tag{12}$$

For de-noising the news text data, we follow the method proposed by Zhao et al. (2021). Specifically, we selects news provided by Reuters that is labeled with the "RIC" tag and filters for news related to the stocks in TOPIX500 based on their stock identifiers, ensuring only relevant news is extracted.

3.1.2 Compared Methods

To evaluate the effectiveness of LED-GNN, we compared it with a range of baseline models, including traditional machine learning methods, classical and state-of-the-art time series prediction models, and dynamic graph neural networks.

Traditional methods used in stock prediction: **Exponential Moving Average (EMA)** (Holt, 2004), a variant of Moving Average; **Linear Regression (LR)** (Galton, 1886).

Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a classical time series forecast method and is widely applied in stock market prediction tasks.

Temporal Convolutional Networks (TCN) (Bai et al., 2018) uses causal and dilated convolutions for time series prediction.

TimesNet (Wu et al., 2023a) is a SOTA model for time series prediction that transforms one-dimensional time series into 2D tensors, capturing intra- and inter-period variations for complex temporal patterns.

SegRNN (Lin et al., 2023) is an novel RNN architecture that improves forecasting through segmented iterations and Parallel Multi-step Forecasting (PMF).

MTGNN (Wu et al., 2020) is a graph-based time series prediction model that capture the dependencies within multivariate time series with graph learning.

RTGCN (Zheng et al., 2023) is a stock prediction model that represents relationships between stocks as a relational temporal graph, utilizing relation-aware strategies for feature extraction.

3.1.3 Implementation and Metrics

Metrics As is mentioned in Section 2.1, we formalize the trading volume prediction as a regression problem. Thus, we select mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and symmetric mean aboslute percentage error (SMAPE) as indicators of the performance of the models.

Experiment Setting All experiments were conducted on a Tesla V100-SXM2-32GB GPU and an Intel Core 2 Duo T7700 processor. All models aside from EMA employ the Adam optimizer with an initial learning rate of 0.001. The learning rate was reduced to one-tenth using the Reduce on Plateau strategy when the loss remained unchanged for 10 consecutive epochs. The mean squared error (MSE) was used as the loss function, and the best result after the loss stabilized was taken as the final result for each model. For the RTGCN model, since the original edge relationships were not available, experiments were conducted using LLM-generated news edges as input. In baseline comparisons and module effectiveness evaluations, a lookback window size of 30 (representing the number of historical time points available to the model) and a batch size of 24 were used.

3.2 Experiment Results

3.2.1 Prediction Performance

Model	MSE	RMSE	MAE	SMAPE
MA	0.608	0.780	0.644	1.882
LinearR	0.171	0.413	0.310	0.729
LSTM	0.157	0.396	0.297	0.699
TCN	0.159	0.398	0.298	0.702
Timesnet	0.164	0.405	0.304	0.712
Segrnn	0.166	0.407	0.305	0.715
MTGNN	0.161	0.401	0.304	0.709
RTGCN	0.160	0.400	0.299	0.704
LED-GNN	**0.153(-2.5%)**	**0.391(-1.3%)**	**0.293(-1.4%)**	**0.680(-2.7%)**

Table 1: Comparison of baseline models. The best results are highlighted in bold, and relative improvements are shown in parentheses.

Table 1 shows that LED-GNN outperforms all baseline models across all metrics (highlighted in bold). Compared to the second-best model, LED-GNN achieves improvements of 2.6%, 1.3%, 1.0%, and 2.8% in MSE, RMSE, MAE, and SMAPE, respectively. Traditional machine learning models like Moving Average and Linear Regression perform the worst, likely due to their simplicity and inability to capture complex relationships in the data.

TimesNet and SegRNN, despite achieving state-of-the-art results on many time-series datasets, underperform in stock volume prediction, likely due

Model	MSE	RMSE	MAE	SMAPE
LED-GNN	**0.153**	**0.391**	**0.294**	**0.680**
w/o news graph	0.155	0.393	0.295	0.691
w/o stock graph	0.155	0.394	0.295	0.692
random graph	0.159	0.396	0.299	0.695

Table 2: Ablation experiments showing the performance of LED-GNN and its variants.

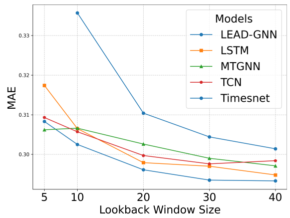
to the high-frequency fluctuations in stock data and their difficulty in handling noise. Their large parameter counts (13x and 4x that of LSTM, respectively) may exacerbate overfitting. In contrast, spatiotemporal graph models like RTGCN and MTGNN outperform these models by capturing stock dependencies and simulating market interactions. The graph-based approach also helps mitigate overfitting by providing structural information that directs the model to focus on key patterns.

Surprisingly, LSTM and TCN perform very well. This could be because they are well-suited to handling the high-frequency fluctuations and noise in stock market data. These two models have fewer parameters and stronger generalization ability, allowing them to better resist noise interference in stock volume prediction. In previous studies on stock volume prediction, LSTM also performed excellently, even surpassing all other baseline models multiple times (Zhao et al., 2021).

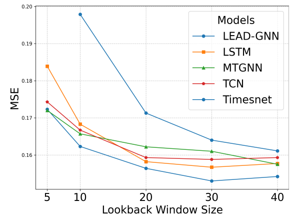
Among all models, LED-GNN achieves top performance by mining dynamic relationships from news using large language models and extracting dependencies between stock time series through end-to-end graph structure learning. Additionally, incorporating external news data helps mitigate overfitting by providing relevant contextual information that guides the model to focus on significant patterns.

3.2.2 Effectiveness of Sub-modules

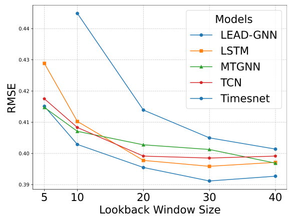
We assess the significance of key sub-modules in our framework through ablation experiments, with results presented in Table 2. The variant w/o news graph excludes the dynamic news relation graph derived from news, while w/o stock graph removes the stock spatiotemporal graph. Additionally, LED-GNN with random graph replaces both the stock spatiotemporal graph and the dynamic news relation graph with a randomly generated graph containing the same number of edges as the original news graph.



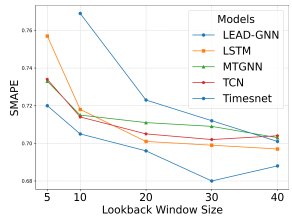
(a) Effect on MAE



(b) Effect on MSE



(c) Effect on RMSE



(d) Effect on SMAPE

Figure 5: Effect of lookback window size on the performance of different models. We select the window size of from 5 to 40.

The complete LED-GNN outperforms the other ablation models. Specifically, the results of w/o news graph and w/o stock graph are slightly worse than the complete model but still outperform all other baseline models, demonstrating the effectiveness of both graphs in the stock volume prediction task. There is no significant performance gap between the two variants, indicating that the accuracy of the LED-GNN model does not depend solely on either graph. Both the news dynamic relationship graph and the stock spatiotemporal graph contribute to the model's ability to capture complex relationships in the stock market. The results of LED-GNN with a random graph are the worst, demonstrating that strong performance is based on a carefully designed structure while random edge information can dilute useful information and and hinder the model's effectiveness.

3.2.3 Lookback Window Selection

The length of the lookback window determines the length of the historical sequence the model can perceive, thereby affecting its prediction performance. The window sizes selected for the experiment are 5, 10, 20, 30, and 40. Since SegRNN and Linear Regression (LinearR) perform poorly, they were excluded from the figure for better visualization of the other models' performance.

As shown in Figure 5 , both LED-GNN and other models show improved performance as the window size increases, but the performance gains diminishes with larger windows. As the lookback window expands, the model can access longer historical sequences, which helps it capture temporal dependencies in the time series more accurately. However, the performance gains slow down due to the potential noise introduced by the increase in window size. Combined with the limited long-term dependence of stock data, metrics show a tendency of stablizing or even decreasing after the window size exceeds 30.

LED-GNN performs well across all window sizes. Compared to LSTM, LED-GNN improves MSE by 5.80%, 3.39%, 0.57%, 2.36%, and 2.22% at window sizes of 5, 10, 20, 30, and 40, respectively. It is clear that, compared to LSTM and other time series models like TCN, LED-GNN demonstrates a stronger advantage with both shorter sequences (less than 20) and longer sequences (greater than or equal to 30).

When handling shorter time series, both MT-GNN and LED-GNN outperform traditional time series models. At a window size of 5, MTGNN's performance is nearly on par with LED-GNN. This is likely because MTGNN and LED-GNN incorporate topological information, which increases the effective sample size and reduces overfitting. This demonstrates that in data-scarce scenarios, introducing graph structure information can lead to good prediction performance.

As the window size increases, the performance gap between LED-GNN and MTGNN widens, likely because LED-GNN's dynamic news relationship graph and stock spatiotemporal graph enable it to better capture interactions between stocks over longer sequences.

4 Related Work

4.1 Graph Neural Network in Stcok Prediction

Compared to traditional time series prediction models, Graph Neural Networks (GNNs) exhibit significant advantages in handling stock time series data by incorporating interstock relations in addition to intrastock information. However, due to the lack of inherent graph structures in the stock data, different techniques are used to construct the graph. Some literature utilizes prior knowledge to construct knowledge graphs or heterogeneous graphs as a foundation for subsequent prediction tasks, using domain knowledge (Sawhney et al., 2021b), company and industry documents (Gao et al., 2021; Hsu et al., 2021), encyclopedia knowledge (Kim et al., 2019; Zheng et al., 2023) and personnel and sector information (Zhao et al., 2022). Others obtain edge information from historical stock price and trading volume, including deriving static graphs from the correlation matrix (Xiang et al., 2022; Zhao et al., 2021), or learning the graph structure in a end-to-end manners (Uddin et al., 2021; Sawhney et al., 2021a; Li et al., 2022).

4.2 Large Language Models in Stock Prediction

The application of large language models in stock market analysis is mostly confined to natural language tasks such as virtual finance assistant and stock movement prediction (Xie et al., 2023; Yang et al., 2023b). These models can be broadly categorized into mixed-domain LLMs and more cost-efficient instruction-finetuned LLMs (Lee et al., 2024). An example of the former is BloombergGPT (Wu et al., 2023b), which is trained on a large general-purpose corpus combined with an extensive financial-specific dataset. In contrast, the latter category includes models like FinMA (Xie et al., 2023), InvestLM (Yang et al., 2023b), and FinGPT (Yang et al., 2023a), which focus on fine-tuning for financial tasks with reduced computational demands.

5 Conclusion

In this work, we introduced the LED-GNN framework, a novel approach to trading volume prediction that integrates dynamic relationship graphs derived from both historical stock data and news articles, enhanced through large language models (LLMs). By modeling news as graph edges, LED-GNN captures the intricate interactions between stocks influenced by external events, offering a more comprehensive representation of stock relationships. Extensive experiments are conducted to evaluate the performance of LED-GNN and the effectiveness of its sub-modules. Additionally, we explore the impact of the lookback window length on prediction accuracy. Our model outperforms all baselines consistently. To the best of our knowledge, this is the first work to apply large language models for extracting stock news to construct dynamic graphs. We hope this work will inspire further exploration of the integration of large language models and graph neural networks in the field of stock prediction.

Acknowledgments

We thank all the anonymous reviewers for their valuable suggestions. This work is supported by a Research Grant from Mizuho Securities Co., Ltd. We sincerely thank Mizuho Securities for valuable domain expert suggestions. Ruihan Bao and Xu Sun are the corresponding authors.

References

Mebarka Allaoui, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. 2020. Considerably improving clustering algorithms using UMAP dimensionality reduction technique: A comparative study. In *Image and Signal Processing*, pages 317–325, Cham. Springer International Publishing.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *Preprint*, arXiv:1803.01271.

Jędrzej Białkowski, Serge Darolles, and Gaëlle Le Fol. 2008. Improving VWAP strategies: A dynamic volume approach. *Journal of Banking & Finance*, 32(9):1709–1722.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *International Conference on Learning Representations*.

Christian T. Brownlees, Fabrizio Cipollini, and Giampiero M. Gallo. 2010. Intra-daily volume modeling and prediction for algorithmic trading. *Journal of Financial Econometrics*, 9(3):489–518.

A. Ronald Gallant, Peter E. Rossi, and George Tauchen. 2015. Stock prices and volume. *The Review of Financial Studies*, 5(2):199–242.

Francis Galton. 1886. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263.

Jianliang Gao, Xiaoting Ying, Cong Xu, Jianxin Wang, Shichao Zhang, and Zhao Li. 2021. Graph-based stock recommendation by time-aware relational attention network. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16:1 – 21.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *Preprint*, arXiv:2203.05794.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Charles C. Holt. 2004. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10.

Yi-Ling Hsu, Yu-Che Tsai, and Cheng-Te Li. 2021. Fingat: Financial graph attention networks for recommending top-k profitable stocks. *Preprint*, arXiv:2106.10159.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arxiv:2311.05232.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *Preprint*, arxiv:1611.01144.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *Preprint*, arxiv:2310.06825.

Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang. 2019. Hats: A hierarchical graph attention network for stock movement prediction. *Preprint*, arXiv:1908.07999.

Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. 2024. A survey of large language models in finance (finllms). *Preprint*, arXiv:2402.02315.

Xiaojie Li, Chaoran Cui, Donglin Cao, Juan Du, and Chunyun Zhang. 2022. Hypergraph-based reinforcement learning for stock portfolio selection. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4028–4032.

Xin Liang, Dawei Cheng, Fangzhou Yang, Yifeng Luo, Weining Qian, and Aoying Zhou. 2020. F-hmtc: Detecting financial events for investment decisions based on neural hierarchical multi-label text classification. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 4490–4496. International Joint Conferences on Artificial Intelligence Organization.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. 2023. SegRNN: Segment recurrent neural network for long-term time series forecasting. *Preprint*, arxiv:2308.11200.

Xiao-Yang Liu, Guoxuan Wang, Hongyang Yang, and Daochen Zha. 2023. Fingpt: Democratizing internet-scale data for financial large language models. *Preprint*, arXiv:2307.10485.

Andrew W. Lo. 2004. The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management*, 30(5):15–29.

Leland McInnes and John Healy. 2017. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE.

Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arxiv:1802.03426.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *Preprint*, arxiv:1908.10084.

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. 2021a. Stock Selection via Spatiotemporal Hypergraph Attention Network: A Learning to Rank Approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):497–504.

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. 2021b. Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading. In *Proceedings of the Web Conference 2021*, page 11–22, New York, NY, USA. Association for Computing Machinery.

Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2016. Structured sequence modeling with graph convolutional recurrent networks. *Preprint*, arxiv:1612.07659.

Chao Shang, Jie Chen, and Jinbo Bi. 2021. Discrete graph structure learning for forecasting multiple time series. *Preprint*, arXiv:2101.06861.

Robert J Shiller. 1987. Investor behavior in the october 1987 stock market crash: Survey evidence. Working Paper 2446, National Bureau of Economic Research.

Ajim Uddin, Xinyuan Tao, and Dantong Yu. 2021. Attention based dynamic graph learning framework for asset pricing. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. *Preprint*, arXiv:1710.10903.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023a. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023b. Bloomberggpt: A large language model for finance. *Preprint*, arXiv:2303.17564.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 753–763, New York, NY, USA. Association for Computing Machinery.

Sheng Xiang, Dawei Cheng, Chencheng Shang, Ying Zhang, and Yuqi Liang. 2022. Temporal and heterogeneous graph neural network for financial time series prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *Preprint*, arXiv:2306.05443.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023a. Fingpt: Open-source financial large language models. *Preprint*, arXiv:2306.06031.

Yi Yang, Yixuan Tang, and Kar Yan Tam. 2023b. Investlm: A large language model for investment using financial domain instruction tuning. *Preprint*, arXiv:2309.13064.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Liang Zhao, Wei Li, Ruihan Bao, Keiko Harimoto, YunfangWu, and Xu Sun. 2021. Long-term, short-term and sudden event: Trading volume movement prediction with graph-based multi-view modeling. *Preprint*, arXiv:2108.11318.

Yu Zhao, Huaming Du, Ying Liu, Shaopeng Wei, Xingyan Chen, Fuzhen Zhuang, Qing Li, Ji Liu, and Gang Kou. 2022. Stock Movement Prediction Based on Bi-Typed Hybrid-Relational Market Knowledge Graph via Dual Attention Networks. *IEEE Transactions on Knowledge and Data Engineering*, 35:8559–8571.

Z. Zheng, J. Shao, J. Zhu, and H. T. Shen. 2023. Relational Temporal Graph Convolutional Networks for Ranking-Based Stock Prediction. In *IEEE 39th International Conference on Data Engineering (ICDE)*, pages 123–136.

Zhihan Zhou, Liqian Ma, and Han Liu. 2021. Trade the event: Corporate events detection for news-based event-driven trading. *Preprint*, arXiv:2105.12825.

Jinan Zou, Haiyao Cao, Lingqiao Liu, Yuhao Lin, Ehsan Abbasnejad, and Javen Qinfeng Shi. 2022. Astock:

A new dataset and automated stock trading based on stock-specific news analyzing model. *Preprint*, arXiv:2206.06606.