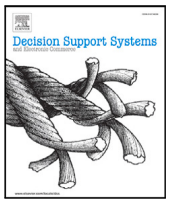
Decision Support Systems 198 (2025) 114528

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss



Modeling hybrid firm relationships with graph neural networks for stock investment decisions

Yang Du [a], Biao Li [a], Zhichen Lu [c], Gang Kou [b,a,*]

[a] School of Business Administration, Southwestern University of Finance and Economics, Chengdu, 611130, China
[b] Xiangjiang Laboratory, Changsha, 410205, China
[c] Research Center on Fictitious Economy and Data Science, the Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing, 100190, China

ARTICLE INFO

Keywords:
Artificial intelligence
Graph neural network
Finance
Stock prediction
Investment decisions

ABSTRACT

The highly volatile nature of the stock market makes predicting data patterns challenging. Significant efforts have been dedicated to modeling complex stock correlations to improve stock return forecasting and support better investor decision-making. Although various predefined intrinsic associations and learned implicit graph structures have been discovered, they have limitations in fully exploring and leveraging both types of graph information. In this paper, we proposed a Hybrid Structure-aware Graph Neural Network (HSGNN) framework. Unlike models that rely solely on predefined or learned graphs, HSGNN utilizes money-flow graphs to complementarily learn implicit graph structures and applies sparse supply-chain graphs to jointly enhance stock return forecasting. Extensive experiments on real stock benchmarks demonstrate our proposed HSGNN outperforms various state-of-the-art forecasting methods, offering a robust decision-support system for financial stakeholders.

1. Introduction

Stock investment is a critical and popular financial decision problem, yet the stock market is a complex and highly volatile system, where firms are interconnected, and various factors interact to cause fluctuations in stock prices. Effective investments are difficult to achieve without a strong ability to predict these movements [1]. Previous studies in stock prediction from finance literature, including traditional asset pricing models and multi-factor models primarily focus on explaining the impact of different factors on the return of individual stock assets [2–4]. In the field of computer science, stock return prediction is typically treated as a time series prediction problem. Various machine learning-based techniques have been proposed to capture the temporal patterns of stock price fluctuations [5–7]. Nevertheless, these approaches typically assume that stocks are independent, often overlooking the influence of correlations between stocks on their expected returns, a phenomenon known in finance as momentum spillover effects [8].

Graph neural networks (GNNs) offer a powerful framework for capturing relational information in graph-structured data [9], leveraging both individual entities's attributes and the intricate network of their interactions to learn rich, informative representations. Given this capability, it is natural to use GNNs to capture momentum spillovers in stocks [10–12] or other financial assets [13]. The entire stock market can be viewed as a graph, with firms as nodes and their relationships as edges, which are used to converge and aggregate momentum.

It is important to note that the quality of graph-structured data is crucial, whether for quantitative analysis or stock prediction modeling using graph deep learning. The inter-firm relationships used for stock prediction can be broadly categorized into two types: predefined graphs, known as *explicit graphs*, and learned graphs, known as *implicit graphs*. In detail, the most widely used explicit graphs are built upon intrinsic firm-level associations, including industry classification, geographic proximity, and supply chain linkages. These graphs, while interpretable, are limited by their reliance on predefined structures. Due to the complexity of the market, the relationships between firms are often intricate and diverse. The explicit graph constructed based solely on a single intrinsic attribute struggles to fully cover all associations. Additionally, relationships between firms evolve over time, continually adapting to shifts in market competition, policy environments, technological advancements, and corporate strategies. For example, BYD (002594.SZ) and CATL (300750.SZ) are key players in the new energy vehicle supply chain. Initially, BYD relied on CATL for batteries, but as BYD developed its own battery capabilities, the partnership weakened and they became competitors in the global power

* Corresponding author at: School of Business Administration, Southwestern University of Finance and Economics, Chengdu, 611130, China.
E-mail address: kougang@swufe.edu.cn (G. Kou).

https://doi.org/10.1016/j.dss.2025.114528
Received 12 January 2025; Received in revised form 22 August 2025; Accepted 1 September 2025
Available online 7 September 2025

battery market. Crucially, explicit graphs are inherently slow-changing: industry and geographic location [14,15] are constant, whereas supply chain relationships and shared analyst coverage [16–18] are slow-changing and typically updated annually, constrained by the frequency of a company's annual report or analyst report updates. This slow adaptation leads to a temporal mismatch: relying solely on a slowly evolving explicit graph is insufficient to capture the complex and dynamic relationships among firms.

To address this limitation, recent approaches have turned to data-driven *implicit graphs*, which infer latent inter-firm dependencies directly from firm-level features such as returns, fundamentals, or multi-dimensional technical indicators [12]. While this allows for better adaptation to market dynamics, such graphs often lack structural priors and can suffer from overfitting to spurious correlations [19]. Therefore, a robust stock prediction model must reconcile two competing demands: adaptability to changing relationships, and stability rooted in economically interpretable structures.

To this end, we propose **HSGNN**, a hybrid graph learning framework that integrates both structural and data-driven perspectives. The model is composed of two key modules:(1) a **structure-aware implicit graph learning** module that constructs a latent graph dynamically, regularized by a behavior-based explicit graph derived from high-frequency money-flow data, and (2) an **explicit graph attention learning** module that models long-term intrinsic associations, supply chain relationships, via an attention mechanism over static, interpretable graphs.

In this design, the implicit graph is not learned in isolation, but is guided by a behavior-driven explicit money-flow graph constructed from high-frequency level-2 trading data. This graph is inspired by the habitat-based comovement theory [20], where correlated stock returns may arise from coordinated trading within investor groups. The money-flow graph quantifies such coordination through synchronized buying and selling behaviors, serving as a domain-informed filter that both validates latent connections and suppresses spurious edges in the implicit graph. Compared to the slow-changing supply chain graph, this rapidly evolving money-flow graph better aligns with the short- to medium-term nature of inter-firm dependencies captured by the implicit graph. In parallel, we incorporate the supply-chain graph as a complementary static view to capture persistent economic linkages. This multi-view design enables HSGNN to simultaneously model long-term structural dependencies and short-term behavioral dynamics in a coherent, interpretable, and effective manner.

Our contributions are summarized as follows:

- We introduce an innovative approach for constructing dynamic money-flow explicit graphs for the subsequent learning process, which leverages the frequency of simultaneous net money inflows or outflows between stocks within the same period.
- We propose a structure-aware graph learning method that leverages money-flow explicit graphs to guide the generation of implicit graphs during the learning process, enhancing implicit stock embedding learning.
- Our study proposes a hybrid graph learning method based on a cross-graph attention mechanism that integrates explicit stock embeddings learned from supply-chain graphs with implicit stock embeddings derived from structure-aware graph learning.
- Extensive experiments with various graph learning models are conducted to evaluate the effectiveness of our proposed method in forecasting stock returns and acquiring portfolio returns. The experiment results also show that our model effectively leverages the incremental benefits provided by graph information.

Our research focuses on the A-share market in China, the second-largest stock market in the world, known for its comprehensive industry and supply-chain distribution among emerging markets. This market offers a wealth of structured data for modeling firm relationships using GNNs [18]. Moreover, as highlighted by [21], the A-share market is primarily driven by retail investors, whose emotional and often irrational behavior creates higher levels of noise and volatility compared to developed markets. This makes the A-share market an ideal setting for GNNs, which are particularly well-suited to model nonlinear relationships and uncover hidden market trends and co-movement patterns, improving predictions of stock price behavior in such a dynamic, non-rational market environment.

2. Related work

2.1. Factors and stock returns prediction

In quantitative trading, it is common practice to convert raw historical stock data, such as open and close prices, into indicators that explain market phenomena, asset returns, and signal market trends. These indicators are known as alpha factors. Some studies have focused on the problems of mining and discovering the factors, a common solution for which is the evolutionary algorithm [22,23], which benefits from global optimization and astringency. Yu et al. [24] used reinforcement learning as an alpha generator to find better formulaic alphas, enhancing the existing set. In industry, WorldQuant [25] presented 101 real-life and explicit quant trading alphas, and empirically proved their effectiveness. Hence, a factor base is essential for constructing investment strategies, providing multi-dimensional data that aids investors in building more effective quantitative models. The Alpha158 factor library in Qlib offers 158 carefully curated financial factors [26], drawn from a range of traditional financial metrics and technical indicators. These factors enable investors to construct predictive models, evaluate stocks, and enhance decision-making with data-driven insights for more effective trading strategies and portfolio management.

2.2. Graphs and momentum spillover effect

Moskowitz and Grinblatt [14] were the first to propose the existence of industry momentum, discovering that past industry returns could predict future stock returns, even after controlling for momentum at the individual stock level. Cohen and Frazzini [16] and Menzly and Ozbas [17] found that past customer returns could predict future returns for supplier companies. Cohen and Lou [27] used segment data to find that the returns of single-segment companies could predict the returns of multi-segment companies operating in the same industry. Parsons et al. [15] identified the presence of regional momentum, showing that the past stock returns of companies headquartered in the same economic region could predict future stock returns, a phenomenon that persists even after controlling for industry momentum. Lee et al. [28] found technological linkages between companies with similar patent portfolios, where the past stock returns of technology-linked companies could predict the future stock returns of target companies.

2.3. Implicit dynamic graph learning

In academia, graph neural networks and deep learning models have been widely adopted for extracting valid information from noise-filled financial data and making contributions to predicting returns. Momentum spillovers are important to investors because they can provide an additional investment strategy. Some works have explored the momentum spillover effects of related companies based on graph neural network models.

As listed in Table 1, Cheng and Li [12] proposed an unmasked attention mechanism to infer the latent relation graph while ignoring all predefined relations. While their experiments showed that fusing the predefined graph did not improve implicit relation inference, their fusion method was constrained to a simple addition of the implicit weighted graph and the predefined unweighted graph, which have different edge attributes. This simplification led to explicit graphs

Table 1
Comparison of our research with existing GNN-based stock prediction research.

Study	Explicit graphs used	Graph type	Methods for implicit graph/ Learned edge weights	Methods of fusion	Graph learning method
Cheng et al. [12]	×	×	Unmasked attention mechanism	×	Attribute-mattered aggregator
Feng et al.[11]	Sector-industry, Wiki company-based	Static and unweighted	Explicit and implicit modeling	×	Enhanced GCN with dynamically learned graph weights
Xu et al. [29]	Industry, business, shareholder, downstream/upstream	Static and unweighted	Masked attention based on stock context encoder	×	Enhanced GCN with dynamically learned graph weights
Gao et al. [30]	Sector-industry	Static and unweighted	Masked attention	×	Enhanced GCN with learned sector-industry weights
Zheng et al. [31]	Sector-industry, Wiki company-based	Static and unweighted	Uniform/Weight/Time-sensitive Strategy	×	Relational Temporal Graph Convolutional Networks
Tian et al. [32]	PSCMG, MSCMG	Daily and unweighted	Hybrid attention encoder	×	HAD-GNN
Chen et al. [10]	Shareholding graph	Static, Weighted	×	×	GCN
Li et al. [33]	Stock correlation graph	Static, Weighted	×	×	RGCN
Song et al. [34]	Industry, Wiki, price similarity	Static and unweighted,	×	Multi-relational graph attention ranking for explicit graphs	GCN
Our Study	Supply chain, money-flow graphs	Annual and unweighted supply chain, monthly and weighted money-flow	Unmasked attention mechanism	Hybrid GNN	GAT for unweighted

being treated as noise. Feng et al. [11] constructed a relational stock ranking framework based on sector-industry relations and wiki relations, including supplier-customer and ownership relations. They fused the temporal evolution and correlation of stocks by improving the component of the graph convolutional network (GCN). The research accounted for dynamics when modeling stock relationships. However, the explicit graphs are represented as unweighted binary-encoded graphs, overlooking the strength of relationships between firms.

Similarly, Xu et al. [29] constructed an industry and shareholder graph and designed a propagation layer to propagate the effect of event information from related stocks; Gao et al. [30] proposed a time-aware relational attention network model that effectively captures time-varying correlation strengths between stocks within sector-industry relations using a time-aware relational attention mechanism, leading to improved stock recommendation performance; Zheng et al. [31] introduced time-sensitive strategies for relational modeling on sector and wiki graphs. All these methods rely on static explicit graphs. In the absence of predefined edge weights, various approaches are employed to learn the strengths of existing relationships. Although the learned edge weights may evolve dynamically, the underlying graph structure remains static—only the connections predefined in the explicit graph are considered during learning.

To model dynamic stock relationships, Tian et al. [32] constructed two types of co-movement graphs based on historical price behavior: the Pearson stock co-movement graph (PSCMG) and the Manhattan stock co-movement graph (MSCMG). These graphs serve as inputs to the hybrid-attention dynamic graph neural network (HAD-GNN) that integrates temporal and node-level attention to capture historical patterns and inter-stock influence jointly. However, the effectiveness of explicit graphs is limited by their unweighted nature.

Chen et al. [10] proposed two strategies to integrate the financial investment graph and show that the related companies' information can better predict the stock price. Li et al. [33] constructed a stock correlation graph based on historical market prices and enhanced the LSTM with the relational graph convolutional network (RGCN) model to predict overnight stock movements by incorporating overnight financial news. Although the graphs incorporate weighted relationships, their static representation and limited relationship types fail to capture the dynamic evolution of market interactions.

Song et al. [34] constructed three explicit graphs — price similarity, Wiki relations, and industry relations — and trained them separately using a two-layer GCN to extract stock relation embeddings. An adaptive attention mechanism is then applied to fuse these embeddings into a unified representation. Despite incorporating various explicit graphs and employing multi-graph fusion, the absence of implicit graphs limits the model's ability to capture evolving relational dynamics.

2.4. Research gaps

Financial markets inherently exhibit a graph structure, where firms are interconnected through complex economic, behavioral, and operational relationships such as supply chains, ownership networks, and industry affiliations. Modeling these markets as graphs enables the explicit representation of such relational dependencies, facilitating more structured and relational reasoning in downstream financial tasks such as stock price prediction, risk propagation, and portfolio optimization.

However, traditional deep learning approaches, such as Deep Neural Networks (DNNs) and Long Short-Term Memory networks (LSTMs), are inherently limited in their ability to capture these interdependencies. DNNs treat each stock as an independent input and excel at learning nonlinear patterns in static features, yet they fail to model temporal dynamics or cross-firm relationships. LSTMs are designed for sequential data and can learn trends over time within individual stocks, but they lack mechanisms to model interactions between different stocks. For instance, if Company A is a major supplier to Company B, and Company B suffers a sudden financial shock, this disruption may significantly impact Company A's stock, a dependency LSTMs cannot effectively represent.

In contrast, Graph Neural Networks (GNNs) naturally encode such inter-firm relationships by modeling stocks as nodes and their connections, whether based on supply chain ties or co-movement statistics, as edges. Through message passing and neighborhood aggregation, GNNs generate node representations that incorporate not only a firm's own features but also those of its connected peers. This structure-aware learning process is better suited to capturing the intricate, dynamic, and interconnected nature of financial markets than traditional deep learning models.

Despite the promise of GNNs, several critical research gaps persist:

- **Isolated Use of Explicit or Implicit Graphs:** Most existing methods utilize either explicit graphs or implicit graphs in isolation. This limits their ability to synergistically combine complementary relational cues, as there are no dynamic fusion mechanisms to integrate these two types of representations effectively.
- **Neglect of Weight Information in Explicit Graphs:** Many implicit graph construction methods rely solely on node-level features and overlook the valuable edge weight information encoded in explicit graphs. This underutilization of domain knowledge can result in reduced model performance and poorer robustness.
- **Limited Attention to Global Dependencies:** Current implicit graph learning approaches often focus on local neighborhoods by calculating edge weights only among explicit neighbors and may overlook long-range interactions. This limitation reduces their ability to capture global market dynamics and leads to potential information loss.

These methodological limitations hinder the comprehensive modeling of firm-level interconnections and ultimately limit the predictive power of existing financial forecasting models. To better capture the complex relationships among firms, we propose a dynamic graph construction method based on a structure-aware global attention mechanism. This approach facilitates message passing between any pair of nodes, enabling the dynamic graph to capture global dependencies more effectively and make full use of the existing graph information. By integrating diverse inter-firm relationships, our method captures the nuanced influence among companies, enabling more accurate stock predictions and deeper insights into market dynamics.

3. Preliminaries

3.1. Problem definition

The target to predict in our study is the numerical return rate. We formulated the τ-step prospective return rate prediction as an inductive node regression task. The future time window τ is set to 10, and the ahead stock return i at time t is defined as Y_i^t.

$$Y_i^t = \frac{p_{t+10}^i}{p_t^i} - 1. \tag{1}$$

where the p is the volume weighted average price(VWAP).

Given the target Y_i^t in day t, we use two types of information for the target firm: firm-specific features X_i^t and relational data R^t.

3.2. Firm features information

We leverage existing financial and operational data along with the Qlib platform to calculate the Alpha158 factor [26], creating a set of features that represent the firm's state on day t. Two types of factor bases make up the firm feature set: one is used for node embedding and the other is applied for learning the dynamic graph. In the node embedding learning process, factors $X_h^t = \{h_1, h_2, \ldots, h_N\}$ were calculated using daily price-related data with daily frequency from 2017 to 2023. Factors were formulaic modalities, such as WorldQuant 101 alpha factors, as follows,

$$-1 * correlation(open, volume, 10) \tag{2}$$

where $correlation(open, volume, 10)$ is the correlation between $open$ price and trading $vloume$ over the past 10 d.

For dynamic implicit graph learning, the feature base consists of ten style factors $X_b^t = \{b_1, b_2, \ldots, b_N\}$ derived from the Barra China Equity Model (CNE5) [35]. Developed by MSCI, this model captures multiple dimensions of risk and return, and is designed to illuminate the dynamics of the China A-Shares Market, thereby aiding institutional investors in their decision-making processes.

3.3. Firm relationships data

We employed two types of graphs to model relationships among firms: the unweighted supply chain graph and the weighted money-flow graph. In all cases, the nodes represent firm entities, and additional information such as Barra risk factors is incorporated as node attributes. Formally, we define a graph as $G = (V, E)$, where V denotes the set of firms and E denotes the set of edges that capture inter-firm relationships. The unweighted supply chain graph G_{sc} reflects business dependencies among firms. It is derived from the CSMAR database, which records the top five customers and suppliers disclosed by publicly listed companies in their annual reports and interim announcements (sourced from the Giant Tide website). This supply chain data is updated annually. The adjacency matrix $A^{ex} \in \mathbb{R}^{N\times N}$ for the directed graph G_{sc} is defined such that $A_{i,j}^{ex} = 1$ if there is a directed edge from node i to node j, and 0 otherwise. Notably, the supply chain graph captures only the closest disclosed relationships and does not reflect the strength of these connections, as firms may not fully disclose all transaction partners.

To complement the slow-changing supply chain graph and achieve more frequent updates, we also construct a dynamic weighted money-flow graph. Money flow has been proven to be an effective market sentiment index for the stock market [36]. The money-flow graphs G_{mf} are generated monthly based on the co-occurrence frequency of different types of money inflow or outflow between stocks over the same period, with edge weights reflecting the strength of these co-movement patterns. The detailed construction procedure for the money-flow graphs is provided in Section 5.1.2

4. Method

In this section, we introduce the architecture of the proposed model HSGNN as depicted in Fig. 1, which comprises five main components: (1) **Data Collection Module:** The datasets include money-flow data, Barra factors, Alpha158 factors, and supply chain graphs. (2) **Explicit Graph Learning Module:** This component leverages firm pairs in the supply chain network to enhance the original factor information for learning. (3) **Structure-Aware Implicit Graph Learning Module:** This module generates weighted graphs derived from money-flow data and implicit relationships learned from Barra factors. (4) **Hybrid Encoding Module:** The hybrid encoder integrates information from both intrinsic and implicit graphs. (5) **Prediction Mapping Module:** The prediction module takes a stock's embedding vector from the hybrid encoder, concatenates it with a factor embedding vector, and generates return rate predictions for the stock.

The overall process is formalized in Algorithm 1.

4.1. Structure-aware implicit graph learning

The relationships among firms change over time [11]. To capture the dynamic and complex relationships between firms, we adopted a data-driven implicit graph learning approach to uncover implicit relationships and latent dependencies between firms. To introduce domain knowledge to guide the graph generation during training, we use the dynamic explicit structure to mask learned relationships between nodes. Previous studies that integrated implicit and explicit graphs primarily utilized static, unweighted, binary-encoded graphs [11,12]. Such static graphs often fall short in capturing the dynamic and highly volatile nature of financial markets, resulting in accumulated noise over time. In contrast, the money-flow graphs in our study are updated monthly, enabling more dynamic and adaptable implicit graph generation. Leveraging both the graph structure and node features, we compute a dense adjacency matrix using a structure-aware global attention mechanism, leading to the construction of a structure-aware implicit graph. We infer the latent dependencies between firms $e_{i,j}^{im}$ using an unmasked attention mechanism inspired by AD-GAT [12],

$$e_{i,j}^{im} = \mathbf{a}^\top (W_1 h_i \parallel W_1 h_j) \tag{3}$$

where h_i, $h_j \in \mathbb{R}^{1\times F_1}$ are node feature vectors derived from Barra factors, $\parallel$ represents the concatenation operation, and $W_1 \in \mathbb{R}^{F_1\times F_1'}$ is shared linear transformation matrix, $\mathbf{a} \in \mathbb{R}^{2F_1'}$ is linear transformation weight vector. The learned attention coefficients $e_{i,j}^{im}$ act as edge weights in the fully connected adjacency matrix A_{im}, allowing the model to represent both cooperative and adversarial relationships among firms. To enhance the reliability of the graph structure and reduce the impact of spurious or noisy connections, we introduce an edge consistency validation mechanism that integrates domain knowledge from the money-flow graph to refine the learned implicit graph. The money-flow graph is represented by a fully connected matrix A_{mf}, where each edge weight $e_{i,j}^{mf} \in (-1, 1)$ reflects investor sentiment co-movement. Given the learned attention matrix A_{im} and the money-flow matrix A_{mf}, we apply a kNN-style sparsification by retaining the top-K strongest connections [37]. Next, we perform a sign consistency check

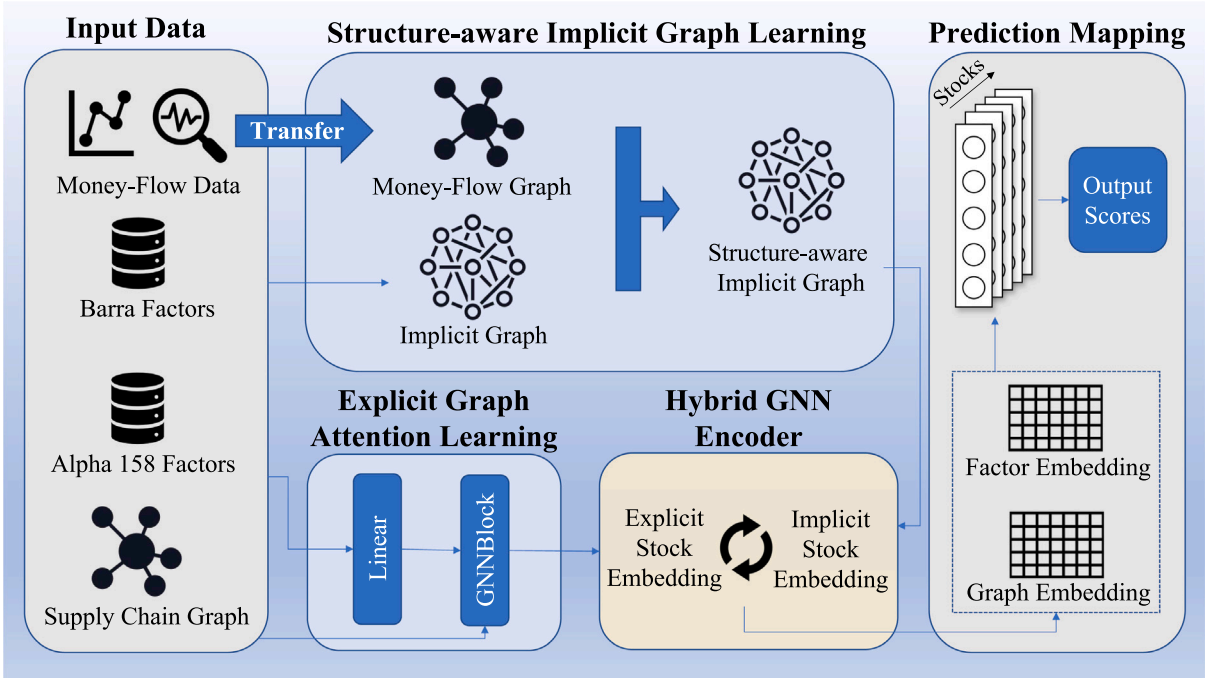


Fig. 1. Overview of the HSGNN Framework. The initial stage constructs both implicit and explicit graphs and encodes the factors integrated with structural information. In the second stage, hybrid encoding adaptively assigns importance to each embedding from the different graph types and aggregates the information accordingly. Then, the prediction mapping stage combines the generated stock embeddings, which incorporate graph structure information, with the factor embeddings. Finally, a fully connected layer is applied to predict each stock's future returns.

between the learned implicit attention weights $e_{i,j}^{im}$ and the explicit edge weights $e_{i,j}^{ex}$, retaining only those edges where the two sources agree in sign. This strategy helps align learned relationships with economically interpretable signals—for instance, preserving edges where both the learned attention and money flow trend indicate positive influence. In contrast, inconsistent edges (e.g., where positive money flow correlation contradicts a negative attention weight) are treated as unreliable and thus discarded. This selective filtering reduces structural noise and improves the robustness of downstream learning. Specifically, the process is as follows:

$$\tilde{A}_{i,j} = \begin{cases} e_{i,j}^{im} \cdot e_{i,j}^{mf}, & \text{if } j \in \mathcal{N}_K(i),\ e_{i,j}^{im} > 0,\ e_{i,j}^{mf} > 0 \\ -e_{i,j}^{im} \cdot e_{i,j}^{mf}, & \text{if } j \in \mathcal{N}_K(i),\ e_{i,j}^{im} < 0,\ e_{i,j}^{mf} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{N}_K(i)$ denotes the set of Top-K neighbors of node i, selected based on the highest absolute values of $e_{i,j}^{im}$ and $e_{i,j}^{mf}$ including self-loops.

Next, because $\tilde{A}$ may contain both positive and negative edge weights, we split it into a "positive submatrix" A^+ and a "negative submatrix" A^- and compute separate attention coefficients over these two sets:

$$A_{i,j}^{+} = \max(\tilde{A}_{i,j}, 0), \quad A_{i,j}^{-} = \max(-\tilde{A}_{i,j}, 0). \tag{5}$$

$$\alpha_{i,j}^{+} = \frac{\exp\left(A_{i,j}^{+}\right)}{\sum_{k:\,\tilde{A}_{i,k}>0} \exp\left(A_{i,k}^{+}\right)}, \quad \alpha_{i,j}^{-} = \frac{\exp\left(A_{i,j}^{-}\right)}{\sum_{k:\,\tilde{A}_{i,k}<0} \exp\left(A_{i,k}^{-}\right)}. \tag{6}$$

We further introduce a sign-aware message passing scheme that explicitly separates the modeling of positive and negative relations. At the first layer, node features are aggregated from neighbors in the positive and negative fused subgraphs independently

$$\begin{aligned} h_i^{+(0)} &= \sigma\left(\sum_{j\in\mathcal{N}_i^+} \alpha_{i,j}^{+} W_0^{+} h_j\right) \\ h_i^{-(0)} &= \sigma\left(\sum_{j\in\mathcal{N}_i^-} \alpha_{i,j}^{-} W_0^{-} h_j\right) \end{aligned} \tag{7}$$

where σ is a non-linear activation function ReLU, $\mathcal{N}_i^+$ and $\mathcal{N}_i^-$ denote the sets of positive and negative neighbors of node i, $W_0^+ \in \mathbb{R}^{d'\times d'}$ and $W_0^- \in \mathbb{R}^{d'\times d'}$ are trainable weight matrices, respectively. In deeper layers ($l \geq 2$), we enable cross-graph propagation by allowing positive (resp. negative) node representations to be updated using both positive and negative neighborhood information from the previous layer:

$$\begin{aligned} h_i^{+(l)} &= \sigma\left(\sum_{j\in\mathcal{N}_i^+} \alpha_{i,j}^{+} W_l^{+} h_j^{+(l-1)} + \sum_{j\in\mathcal{N}_i^-} \alpha_{i,j}^{-} W_l^{-} h_j^{-(l-1)}\right) \\ h_i^{-(l)} &= \sigma\left(\sum_{j\in\mathcal{N}_i^-} \alpha_{i,j}^{-} W_l^{+} h_j^{+(l-1)} + \sum_{j\in\mathcal{N}_i^+} \alpha_{i,j}^{+} W_l^{-} h_j^{-(l-1)}\right) \end{aligned} \tag{8}$$

This cross-sign aggregation scheme allows the model to capture more nuanced structural signals, such as how a firm positively influenced by its allies may still be shaped by opposing dynamics from its competitors. Finally, the outputs from the positive and negative branches are concatenated to form the final node representation:

$$h_i^{im} = [h_i^{+}, h_i^{-}] \tag{9}$$

This dual-path representation captures both collaborative and adversarial influences in the financial network, enabling more expressive and robust modeling of firm-level interactions.

4.2. Explicit graph attention learning

In addition to the learned implicit graphs, we also capture intrinsic associations (such as supply chain relationships in our study) through explicit graph learning. For the publicly available supply chain relationship graph, which is updated annually and relatively static, we employ a Graph Attention Network (GAT) to aggregate the vectors of neighboring nodes in this unweighted explicit graph. Although the supply chain graph lacks explicit edge weights, the supplier-customer relationships exhibit heterogeneity—driven by node-level factors like financial stability and market influence. GAT's attention mechanism

adaptively assigns weights to neighbors based on feature compatibility, allowing the model to focus on more relevant suppliers during aggregation. In contrast, other GNN variants, such as GraphSAGE or GCN, apply fixed or uniform neighbor aggregation schemes, which may fail to capture such nuanced dependencies.

$$e_{ij}^{ex} = \mathbf{a}^{\top} LeakyReLU(W v_i \parallel W v_j) \tag{10}$$

$$\alpha_{ij}^{ex} = \frac{\exp(e_{ij}^{ex})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{ex})} \tag{11}$$

where v_i, $v_j \in \mathbb{R}^{1 \times F_2}$ are node vectors derived from Alpha158 factors. The attention coefficient α is normalized by the softmax function.

Then the explicit relational node embedding vectors $\mathbf{h_{ex}} = \{h_1^{ex}, h_2^{ex}, \ldots, h_N^{ex}\}$ are generated by a linear combination of neighbor node vectors as follows, where the weight is α_{ij}^{ex}.

$$h_i^{ex} = \sigma\Big(\sum_{j \in N(i)} \alpha_{ij}^{ex} W v_j\Big) \tag{12}$$

4.3. Hybrid GNN encoder

While the dynamic implicit graph — learned through attention mechanisms and guided by money-flow signals — effectively captures short-term, sentiment-driven interactions among stocks, it lacks the capacity to represent fundamental and persistent relationships that characterize long-term business dependencies. To address this limitation, we propose a Hybrid GNN Encoder that independently models the slow-evolving explicit supply-chain graph and the dynamic implicit graph. Given the mismatch in update frequency and information decay rates between the two graphs, we adopt a modular modeling strategy: each graph is encoded separately using a dedicated GNN module. Their respective node embeddings are then fused at the representation level. This design preserves the distinct temporal dynamics and financial semantics of each graph while allowing their complementary information to be jointly exploited in downstream prediction tasks. Specifically, a cross-graph attention module is introduced to explicitly compute interactions between node representations from the two graphs. Let h_{im} and h_{ex} denote the representations from the implicit and explicit graphs, respectively. The attention weight is computed as:

$$\alpha = \sigma_1\left(a^{\top} \cdot \tanh\left(W_1 \mathbf{h}_{im} + W_2 \mathbf{h}_{ex}\right)\right) \tag{13}$$

where $\odot$ is the component-wise multiplication, activation function σ_1 is softmax function, W_1 and W_2 is learnable weight matrix and b_r is learnable vector.

Given the static and dynamic aggregated vectors, the fusion function is designed as a gated sum of two inputs, resulting in the output $\mathbf{v_g}$. The fused representation is obtained by:

$$\mathbf{v_g} = Fuse(\mathbf{h}_{im}, \mathbf{h}_{ex}) = \alpha \odot \mathbf{h}_{im} + (1 - \alpha) \odot \mathbf{h}_{ex} \tag{14}$$

This approach projects both representations into a shared interaction space using W_1 and W_2, and dynamically captures their dependencies through the attention mechanism. The use of a tanh activation followed by a softmax-based attention vector introduces a two-layer non-linear mapping, enhancing the model's ability to capture complex cross-graph interaction patterns.

4.4. Prediction mapping

For stock i, its vector representation $\mathbf{v_{fc}}$ is generated by a fully connected (FC) neural network. The input for the prediction module is the concatenation of $\mathbf{v_g}$ and $\mathbf{v_{fc}}$. Finally, a single-layer feed-forward neural network with the sigmoid function is applied to generate the predictions of future stock returns, denoted as

$$\hat{y}_i^t = sigmoid(W_i[\mathbf{v}_{fc} \parallel \mathbf{v}_g] + b_i) \tag{15}$$

where W_i is the matrix of learnable parameters and b_i is learnable vector. The Mean Squared Error (MSE) loss between $\hat{y}_i^t$ and y_i^t is back-propagated to learn the parameters of the proposed framework.

$$loss_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{16}$$

5. Experiments

In this section, we describe the experiment setting, results, and detailed analysis.

5.1. Data

5.1.1. Factors

To predict cross-sectional stock returns, a dataset was constructed based on the Chinese stock market. Daily price-related data and formulaic factors were collected from the Qlib platform from 2017 to 2023. For explicit graph learning, the factors used are Alpha158 factors including a set of 158-dimensional features. For implicit graph learning, style factors derived from the CNE5 model are used to capture various dimensions of risk and return, including beta, book-to-price, earnings yield, growth, leverage, liquidity, momentum, residual volatility, size, and non-linear size.

5.1.2. Graph construction

Supply Chain Graph For explicit graph learning, we collect supply chain network data from CSMAR, which is updated annually. Edge numbers of annual supply chain graphs are not static, specially, customer graphs from 107 to 141, and supplier graphs from 88 to 135. The graph is unweighted and sparse. The construction method is straightforward: the adjacency matrix A^{ex} for a graph with N nodes is an $N \times N$ matrix, where each element $A_{i,j}^{ex}$ equals 1 if there is a directed edge from node i to node j, and 0 if there is no such edge.

Money Flow Graph Most existing public inter-enterprise graphs are static or updated only once a year due to the low frequency of data disclosure. To enhance the robustness of implicit graph learning and better align with the dynamic nature of the stock market, we introduce dynamic money-flow graphs that incorporate domain expertise to capture behaviorally driven comovement in equity markets. Specifically, the money-flow graph is inspired by the habitat-based comovement theory proposed by Barberis et al. [20], which suggests that stock return correlations can arise from coordinated trading activity triggered by shifts in investor sentiment or risk preferences within distinct investor groups, or "habitats". Constructed using high-frequency level-2 market data, our money-flow graphs quantify trading patterns across various investor cohorts and reveal a comovement mechanism driven by collective trading dynamics rather than intrinsic relationships.

To capture behavioral trend similarities among investors, we construct dynamic money-flow correlation graphs using level-2 trading data. The Money Flow Net (MFN) captures daily money-flow movement and is defined as the net value of active buying minus active selling transactions. A positive MFN indicates a net inflow, reflecting market optimism, while a negative MFN signals a net outflow, suggesting market pessimism. To better distinguish the behaviors of different investor cohorts, we further classify MFN according to the size of each transaction. Customer orders are classified into four categories based on the transaction amounts: small orders (retail, $< 40,000$ RMB), medium orders (intermediate, $40,000-200,000$ RMB), large orders (institutional, $200,000 - 1,000,000$ RMB), and extra-large orders (institutional, $> 1,000,000$ RMB). Based on these categories, we construct three MFN variants that represent different market participant groups: MFN_{Cash}, which includes all active orders regardless of size; $MFN_{MedSmall}$, covering only medium and small active orders (under 200,000 RMB), largely reflecting retail investor activity; and $MFN_{ExNLarge}$, consisting of large and extra-large active orders (over 200,000 RMB), typically

Table 2
Statistics of explicit graphs. The supplier and customer graph data are sourced from the CSMAR database. The money-flow graphs are constructed based on fundamental data derived from level-2 high-frequency market data.

	Frequency	Type	Weights	Nodes	Edges	Edges sparsity
Supplier Graph	Annual	Unweighted	{0,1}	3945	98	0.0013%
Customer Graph	Annual	Unweighted	{0,1}	3945	110	0.0015%
Money Flow Graph	Monthly	Complete and Weighted	[−1,1]	3945	7 779 540	1

originating from institutional investors. Here, the focus on active orders is motivated by their immediacy and responsiveness to market sentiment [38]. Active orders, such as market orders or aggressively priced limit orders, reflect deliberate and timely decisions by investors who seek to capture short-term advantages or react to information shocks. Compared with passive orders, which tend to follow slower and rule-based strategies, active orders are better suited to revealing investors' real-time behavioral trends.

Based on the daily values of MFN, we extract binary features to indicate whether a given stock experiences net inflow or outflow on each trading day. For stock i on day t, the indicator is set to 1 if the corresponding MFN satisfies the condition and 0 otherwise. These daily indicators are aggregated monthly to form time series that capture money-flow patterns for each stock and investor group. Thus, for company i in month m with T_m trading days, we define binary time series vectors:

$$\begin{aligned}
&\mathbf{CashIn}_i^m = \left(\mathbf{1}_{\{\mathrm{MFN}_{\mathrm{Cash}}^t>0\}}\right)_{t=1}^{T_m}, && \mathbf{MedSmallIn}_i^m = \left(\mathbf{1}_{\{\mathrm{MFN}_{\mathrm{MedSmall}}^t>0\}}\right)_{t=1}^{T_m},\\
&\mathbf{ExNLargeIn}_i^m = \left(\mathbf{1}_{\{\mathrm{MFN}_{\mathrm{ExNLarge}}^t>0\}}\right)_{t=1}^{T_m}, && \mathbf{CashOut}_i^m = \left(\mathbf{1}_{\{\mathrm{MFN}_{\mathrm{Cash}}^t<0\}}\right)_{t=1}^{T_m},\\
&\mathbf{MedSmallOut}_i^m = \left(\mathbf{1}_{\{\mathrm{MFN}_{\mathrm{MedSmall}}^t<0\}}\right)_{t=1}^{T_m}, && \mathbf{ExNLargeOut}_i^m = \left(\mathbf{1}_{\{\mathrm{MFN}_{\mathrm{ExNLarge}}^t<0\}}\right)_{t=1}^{T_m}.
\end{aligned} \tag{17}$$

Here, $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function, and the vector length T_m corresponds to the number of trading days in month m, typically around 22.

For each flow type $k \in \mathcal{K} = \{\text{CashIn, CashOut, MedSmallIn, MedSmallOut, ExNLargeIn, ExNLargeOut}\}$, we construct monthly correlation matrices:

$$\mathbf{M}_k^m = \left[\rho\left(\mathbf{k}_i^m, \mathbf{k}_j^m\right)\right]_{N\times N} \tag{18}$$

where $\rho(\cdot)$ is Pearson correlation, N is number of stocks, and matrix entries represent behavioral similarity.

The final adjacency matrix combines all money-flow perspectives:

$$\bar{\mathbf{A}}^m = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\mathbf{M}_k^m = \frac{1}{6}\sum_{k\in\mathcal{K}}\mathbf{M}_k^m \tag{19}$$

yields a complete weighted graph $G_{mf} = (V, \bar{\mathbf{A}}^m)$, where nodes V represent stocks and edge weights $\bar{a}_{ij}^m \in [-1, 1]$ reflect money-flow correlation strengths.

Compared to static and sparse supplier-customer graphs, the money-flow graph offers dense, weighted, and frequently updated information that better aligns with the dynamic nature of financial markets. Its continuous edge weights capture real-time sentiment co-movements, enabling more effective guidance for implicit graph learning through masking and edge validation. Moreover, unlike supply-chain relations, money-flow correlations are directly relevant to short-term market behavior and latent risk transmission, making them a more suitable supervisory signal. Table 2 provides a comparison of explicit graph structures, including the supplier, customer and money-flow graphs.

To further validate the relationship between supply chain interactions, money-flow graphs, and stock prices, we conducted an additional statistical experiment. Recognizing that stock price correlations among related stocks are higher than those across the entire market [39], we analyzed daily price data from 2017 to 2023. Specifically, we calculated monthly correlation coefficients as follows, using daily VWAP (Volume Weighted Average Price) for stocks across the entire A-share market ($\overline{corr}_{all}$), as well as for those within the established supply chain ($\overline{corr}_{sc}$) or money-flow relationships ($\overline{corr}_{mf}$).

$$\overline{corr} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{corr}(p_{i,t}, p_{j,t}) \tag{20}$$

where $p_{i,t}$ and $p_{j,t}$ represent the daily VWAP price sequences for stocks i and j during month t. In the case of $\overline{corr}_{all}$, n denotes the total number of stocks. For $\overline{corr}_{sc}$ and $\overline{corr}_{mf}$, n refers to the number of stocks within the supply-chain or money-flow relationships. T represents the total number of months in the time period.

The mean values of the statistical results are presented in Table 3, which reveals differences in the mean correlation coefficients across various relationship networks. The average correlation for all stocks in the A-share market is 27.15%, serving as a baseline. Stocks within the supplier and customer networks show higher correlations, at 34.01% and 32.20%, respectively, indicating stronger price linkage among stocks with supply chain relationships. Money flow networks are constructed as complete graphs. To ensure fair comparison, we retained only the top-k edges with the highest weights, where k matches the number of edges in the supply chain network. Notably, stocks within money flow networks exhibit the highest correlation, at 39.85%, suggesting that money flow relationships may be a more significant indicator of stock price co-movement.

6. Experiments setup

This study aimed to model a cross-sectional stock selection scenario. The returns calculated using the VWAP of day T+11 over day T+1 were regarded as labels for the samples on day T.

A rolling approach was adopted for the training: the model was trained every 10 trading days and applied to stock selection and portfolio construction for 10 trading days in the future. The stock data for the entire market over the past 300 trading days were adopted as the training sample. To ensure temporal alignment, two types of graphs are used according to their respective characteristics. The supply-chain graph, constructed from low-frequency annual data, remains static within each calendar year—training samples in year Y all use the graph from year $Y - -1$. This provides structural consistency and avoids data leakage. In contrast, the money-flow graph, built monthly from high-frequency trading data, captures short-term market sentiment. Each training sample uses the graph from the previous month to reflect recent dynamics without introducing look-ahead bias.

6.1. Model training

6.1.1. Baselines

To show the performance of our proposed model, we compare HSGNN with SOTA methods. We select the following models as the baseline for comparison: (1) 3-layer MLP (Multi-Layer Perceptron);

Table 3
Correlation coefficients statistics among stocks within relationship networks. A higher value indicates a stronger price correlation between the stocks within the corresponding relationship.

	Mean of correlation coefficient
All Stocks	27.15%
Stocks within Supplier Graphs	34.01%
Stocks within Customer Graphs	32.20%
Stocks within Money Flow Graphs(0.0028%)	39.85%

Algorithm 1 Hybrid Structure-aware Graph Neural Network (HSGNN)

Input: $G_{mf} = (V, E_{mf})$: Money flow graph (monthly, dense);
$G_{sc} = (V, E_{sc})$: Supply chain graph (annual, sparse);
X_{barra}: Node features (Barra factors); X_{alpha}: Node features (Alpha factors);
$W_0, W_1, W_2, W, W_r, b_r, W_i, b_i$: Learnable weights and biases
Output: $\hat{y}$: Predicted future stock returns

Step 1: Node Embedding via Implicit Graph Learning
$H_1 \leftarrow X_{barra}W_1$; $H_2 \leftarrow X_{alpha}W_2$ ▷ Linear transformation of node features
$e_{ij}^{im} \leftarrow \alpha(H_1[i], H_1[j])$ ▷ Compute implicit graph weight
$S_{im}, S_{mf} \leftarrow$ Top-K neighbors by $|e_{ij}^{im}|, |e_{ij}^{mf}|$ ▷ Select structurally aligned strong edges
for each edge $(i, j) \in S_{im}, S_{mf}$ **do**
    $\tilde{A}_{ij} \leftarrow$ Sign-aware fusion of e_{ij}^{im}, e_{ij}^{mf} ▷ Keep only sign-consistent edges
end for
$A^+, A^- \leftarrow \max(\tilde{A}, 0), \max(-\tilde{A}, 0)$ ▷ Split into positive and negative subgraphs
$\alpha^+, \alpha^- \leftarrow$ softmax over A^+, A^- ▷ Normalize attention coefficients separately
$h_i^{+(0)}, h_i^{-(0)} \leftarrow \sigma(\sum \alpha^{\pm} W_0^{\pm} H_2[j])$ ▷ Initial aggregation from A^+ and A^-
for $l = 1$ to L **do**
    $h_i^{+(l)} \leftarrow \sigma\left(\sum \alpha^{+} W_l^{+} h_j^{+(l-1)} + \sum \alpha^{-} W_l^{-} h_j^{-(l-1)}\right)$
    $h_i^{-(l)} \leftarrow \sigma\left(\sum \alpha^{-} W_l^{+} h_j^{+(l-1)} + \sum \alpha^{+} W_l^{-} h_j^{-(l-1)}\right)$ ▷ Cross-graph message passing across signs
end for
$h_i^{im} \leftarrow [h_i^{+(L)}, h_i^{-(L)}]$ ▷ Final node representation with dual-path fusion
Step 2: Node Embedding via Explicit Graph Learning
$H \leftarrow X_{alpha}W$
for each edge $(i, j) \in E$ **do**
    $e_{ij}^{ex} \leftarrow \alpha(H[i], H[j])$ ▷ Compute attention score
end for
$a_{ij}^{ex} \leftarrow \text{softmax}(e_{ij}^{ex})$; $h_i^{ex} \leftarrow \sum_{j \in \text{neighbors}(i)} a_{ij}^{ex} H[j]$
Step 3: Hybrid GNN Encoder
$v_g \leftarrow Fuse(h_i^{im}, h_i^{ex}, W_r, b_r)$ ▷ Compute fusion embedding
Step 4: Prediction Mapping
$\hat{y}_i \leftarrow \text{MLP}(v_g, X_{alpha}, W_i, b_i)$ ▷ Predict returns
return $\hat{y}$

(2)Traditional time series modeling methods, including LSTM (Long Short-Term Memory) [40], GRU (Gated Recurrent Unit) [41]. (3) Graph methods, GCN (Graph Convolutional Network) [42]. (4) Some other graph neural network-based implicit graph learning methods, which contain attention-based similarity L-GCN (Location-aware Graph Convolutional Networks) [43], cosine-based similarity IDGL (Iterative Deep Graph Learning) [44], kernel similarity-based AGCN (Adaptive Graph Convolution Network) [45], (5) Implicit graph learning method incorporating the prior initial graph GLCN (Graph Learning-Convolutional Network) [46] (6) Attention-based graph fusion method HAD-GNN (Hybrid Attention Dynamic Graph Neural Network) [32].

6.1.2. Experiment setting

Experiments are conducted on an Ubuntu machine equipped with one Intel(R) Core(TM) i7-13700KF with 16 physical cores. The GPU device is NVIDIA TITAN Xp with 12 GB memory. All models are built by PyTorch. To ensure that all models receive effective training, we train each for 200 epochs and implement an early stopping strategy. For hyperparameters of HSGNN, the learning rate lr and edge filter k are tuned from [0.001, 0.01] and [10, 20, 30, 40], respectively. The final selection is lr=0.001 and k=30 for all datasets. Each experiment was repeated three times, and the average performance was reported.

6.2. Evaluation metrics

Information coefficient (IC), Information Coefficient Information Ratio (ICIR), Rank Information Coefficient (RankIC) and Excess Annualized Return (AR) are commonly used performance metrics in quantitative finance, particularly for evaluating stock selection strategies or multi-factor models [47–49].

IC denotes the correlation coefficient between predicted values and actual returns. The larger the absolute value of IC, the more effective predictive the factors. IC is calculated through the linear correlation between predicted values and returns for all stocks on each day, as follows.

$$IC_t = correlation(f_t, r_t) \tag{21}$$

where f_t is the predicted value at t; and r is the equity return at $t + 10$.

ICIR is the ratio of the mean to the standard deviation of the IC; ICIR considers both the stock selection ability of the factor (represented

Table 4
Comparison results on stock metrics. The methods for comparison are mainly divided into three types: Comparison results on stock metrics (measured by t-test with p-value less than 0.01). MLP-based methods, time series modeling methods, and other graph-based methods. Bold shows the best results and underscore indicates sub-optimal results.

	CSI All Share Index			CSI 1000			CSI 500			CSI 300		
Methods	IC	ICIR	RankIC	IC	ICIR	RankIC	IC	ICIR	RankIC	IC	ICIR	RankIC
MLP	11.78%	1.11	10.25%	10.78%	0.97	9.73%	8.48%	0.63	7.50%	4.70%	0.31	3.67%
LSTM	−0.03%	−0.02	0.04%	−0.27%	−0.08	−0.14%	−0.51%	−0.11	−0.51%	−0.85%	−0.14	−0.76%
GRU	0.42%	0.06	0.36%	0.57%	0.07	0.51%	−0.07%	−0.01	−0.18%	0.10%	0.01	−0.03%
GCN	9.37%	1.00	8.33%	9.11%	0.95	8.37%	6.82%	0.57	6.18%	3.82%	0.26	3.96%
L-GCN	12.19%	1.19	10.39%	11.11%	1.05	9.90%	8.47%	0.67	7.09%	4.76%	0.33	3.90%
IDGL	11.68%	1.06	9.83%	10.49%	0.92	9.22%	7.79%	0.58	6.51%	4.49%	0.29	3.45%
AGCN	11.99%	1.12	10.22%	10.96%	1.00	9.75%	8.23%	0.64	6.92%	4.76%	0.32	4.17%
GLCN	12.02%	1.19	10.19%	10.85%	1.01	9.73%	8.36%	0.67	7.17%	5.51%	0.38	4.53%
HAD-GNN	12.52%	1.22	10.65%	11.30%	1.07	10.13%	8.92%	0.71	7.58%	5.07%	0.36	4.20%
HSGNN (customer)	13.71%	1.38	11.97%	12.34%	1.18	11.23%	10.00%	0.80	8.84%	7.02%	0.49	6.34%
HSGNN (supplier)	**14.31%**	**1.45**	**12.43%**	**12.81%**	**1.25**	**11.67%**	**10.40%**	**0.84**	**9.11%**	**7.25%**	**0.52**	**6.39%**

by IC) and the stability of the stock selection ability of the factor (represented by the reciprocal of the standard deviation of IC). The larger the ICIR, the stronger the correlations between the predicted and the actual returns (higher IC), or the relatively stable correlations (lower standard deviation of IC). ICIR is formulated as follows.

$$ICIR = \frac{\overline{IC}}{std(IC)} \tag{22}$$

RankIC is calculated as the Spearman rank correlation coefficient between the predicted stock rankings and their realized rankings based on returns.

AR is the difference between the annualized return of a portfolio or investment and the annualized return of a benchmark index or risk-free rate over the same period.

6.3. Experiments results and analysis

6.3.1. Performance of the stock selection

Table 4 presents the performance of different methods on the CSI All Share Index, CSI 1000, CSI 500, and CSI 300, with metrics including IC, ICIR, and RankIC. Different indices sets represent stock groups of different sizes and characteristics, enabling a detailed analysis of the model's effectiveness across large-cap, mid-cap, and small-cap stocks. The methods are primarily categorized into three types: MLP-based methods utilizing fully connected networks, time series methods based on recurrent neural networks (including LSTM and GRU), and graph-based approaches using graph neural networks (including GCN, L-GCN, IDGL, AGCN, GLCN, and HAD-GNN). The following is an analysis of the experimental results:

Graph-based methods generally outperform MLP-based and time series methods across most indices, especially in mid-cap and small-cap stocks. Among all models, HSGNN demonstrates the strongest performance, with the version using supplier relationship graphs achieving the highest IC, ICIR, and RankIC across all indices. The customer graph variant also shows strong results, ranking closely behind. This highlights the advantage of leveraging graph structures to capture intricate inter-stock relationships, further validating its effectiveness in learning stock price co-movement.

6.3.2. MLP-based methods

Non-graph-based methods include MLP-based approaches and time series methods. MLP demonstrates stable but moderate performance across various indices. For example, on the CSI All Share Index, the IC reaches 11.78%, and the ICIR is 1.11, establishing it as a competitive baseline. The simplicity of the MLP structure allows it to deliver consistent results across different indices. However, its inability to model complex interdependencies between stocks leads to weaker performance compared to graph-based methods. For instance, its RankIC of 10.25% on the CSI All Share Index is significantly lower than the 12.43% achieved by HSGNN. This performance gap widens further on smaller-cap indices such as CSI 300, where MLP's RankIC drops to only 3.67% compared to HSGNN's 6.39%. This trend suggests that while MLP exhibits moderate predictive power, it has significant limitations in capturing the complex dynamic relationships between stocks.

6.3.3. Time series methods

The results for LSTM and GRU indicate that both models struggle to capture meaningful patterns in stock data across all indices. LSTM exhibits consistently poor performance, with negative IC values across all indices. It records the lowest performance on the CSI 300 with an IC of −0.85%, suggesting its inability to effectively handle the noisy and sparse nature of stock data. On the CSI 1000 and CSI 500, LSTM shows marginally better results but still yields low IC values, indicating limited predictive power.

In comparison, GRU performs slightly better than LSTM but still underperforms relative to other methods. It achieves an IC of 0.42% on the CSI All Share Index, 0.57% on the CSI 1000, and 0.10% on the CSI 300, demonstrating modest improvements over LSTM. However, GRU still struggles with weak inter-stock dependencies and temporal patterns, leading to inconsistent results across different indices.

Both LSTM and GRU face challenges in capturing the complex dynamics of stock prices due to the high volatility and non-linear nature of the market. These time-series models struggle with the inherent noise and weak temporal dependencies in financial data, leading to inconsistent and suboptimal results.

6.3.4. Graph-based methods

Some graph neural network-based implicit graph learning methods (L-GCN, IDGL, and AGCN) rely on similarity measurements but lack explicit structural information. In this study, we retained only the implicit graph learning modules from these models, maintaining a consistent framework structure throughout.

Among graph-based methods, L-GCN, IDGL and AGCN, which use implicit graph learning techniques, show promising results but still fall short of HSGNN. As shown in Table 4, similarity-focused methods such as L-GCN, which do not incorporate explicit structures, underperform compared to HSGNN. Notably, HSGNN outperforms both structure-aware methods like GLCN and HAD-GNN across all A-share stocks. On the CSI All Share Index, HSGNN with supplier graph achieves the highest IC of 14.31%, compared to HAD-GNN (12.52%) and GLCN (12.02%). This demonstrates HSGNN's superior ability to capture complex stock co-movements, especially when integrating both explicit and implicit graph structures, and highlights its effectiveness over other graph-based approaches in modeling the dynamic relationships in financial data.

The results also suggest that explicit relationships play a crucial role in model accuracy. The supplier relationship graph in HSGNN achieves the highest average IC of 14.31% on the CSI All Share Index, outperforming other methods and highlighting the predictive value of structural supply chain data. In contrast, the customer relationship graph in HSGNN yields suboptimal performance, with an average IC of 13.71%, indicating that while customer relationships are valuable, they do not offer the same level of predictive power as supplier relationships.

HSGNN's superior performance across all indices stems from its integration of explicit supply chain relationships and implicit graph learning. Relying solely on implicit similarity (e.g., in L-GCN) results in suboptimal performance, as it misses important domain-specific relationships between stocks. In contrast, HSGNN combines explicit and implicit graph learning, capturing both stock-to-stock dependencies and structural supply chain connections. This enables HSGNN to better capture the underlying patterns in stock movements, leading to more accurate predictions.

Traditional GCN achieves an IC and ICIR of 9.37% and 1.00 in the CSI All Share Index, as well as 9.11% and 0.95 in the CSI 1000, illustrating its strength in capturing fundamental stock interrelations. However, its performance slightly lags behind more advanced models, especially on smaller indices like the CSI 300, where it records a lower IC of 3.82%.

Among the other implicit graph learning methods, L-GCN achieves an IC of 12.19%, slightly outperforming IDGL and AGCN, showcasing its capacity to adapt graph structures and improve stock prediction.

HAD-GNN employs an attention-based graph fusion approach that integrates both explicit and implicit graphs, achieving exceptional performance across all indices. Particularly on the CSI All Share Index, it records an IC of 12.52% and an ICIR of 1.22. This method significantly outperforms GCN and IDGL, with a higher ICIR, demonstrating its robust ability to learn dynamic relationships in stock price movements. GLCN also integrates supply chain structural graph knowledge, enhancing its ability to capture stock co-movements. It achieves an IC of 12.02% on the CSI All Share Index, positioning it as a strong competitor to both L-GCN and AGCN.

Finally, HSGNN demonstrates high performance across all CSI indices, highlighting the effectiveness of its complex graph aggregation and supply chain modeling techniques. This consistency across indices of different market caps underscores HSGNN's ability to model diverse stock interactions effectively.

6.3.5. Performance differences across CSI indices

The models' performances varied across different CSI indices, reflecting the unique characteristics of each index:

HSGNN (supplier) emerged as the top performer across all indices, showcasing its ability to effectively capture complex stock interactions and market dynamics, particularly in the CSI All Share and CSI 1000 indices. HAD-GNN and HSGNN (customer) also demonstrated strong adaptability across all market caps, consistently outperforming traditional models such as MLP, LSTM, and GRU. These traditional models struggle to handle the volatility and noise inherent in financial data, highlighting the superior suitability of graph-based models for stock prediction tasks.

The CSI 1000 and CSI 500 indices, consisting primarily of small- and mid-cap stocks, exhibit higher volatility and are more strongly influenced by dynamic market correlations. These conditions make them ideal environments for graph-based methods like HAD-GNN, GLCN, and HSGNN (supplier), which excel at capturing complex relationships and enhancing prediction accuracy.

The CSI 300, representing large-cap, high-quality stocks, is more stable and less volatile compared to the other indices. While graph-based methods still perform well in this index, their IC and ICIR values are relatively lower. This suggests that graph-based models provide more limited returns for large-cap stocks, whose lower volatility and more linear interrelationships are better suited to traditional factor models.

In summary, the experimental results underscore the advantages of dynamic implicit correlation modeling through graph neural networks (GNNs) in capturing the intricate dynamics of stock markets. Graph-based models like HSGNN (supplier), HSGNN (customer), and HAD-GNN significantly enhance predictive accuracy, particularly for small-cap indices like the CSI 1000, where volatile market conditions and complex stock relationships make them highly effective.

6.4. Backtests

We further evaluate and compare the model outputs using a TopN portfolio strategy [11], which selects the top 10% of stocks based on predicted returns and constructs a portfolio targeting these high-performing stocks. The excess annualized returns of the portfolio are reported in Table 5. Specifically, the stock pool consists of the CSI All Share Index components (excluding stocks under special treatment (ST) and those that are suspended). The backtest period is from June 1, 2017, to May 30, 2023. Stocks are ranked by predicted values and divided into 10 groups, with market cap weighting and rebalancing every 10 days. The transaction fee is 0.15%. To mitigate the influence of calendar effects and path dependency, where asset prices may display predictable patterns tied to specific calendar dates or periods (such as days of the week, months, seasons, or holidays), which could distort results depending on the backtest start date, we divide the investment funds for each group into 10 equal portions. Each portion is independently backtested with a different start date, ranging from T+0 to T+9, resulting in 10 distinct sub-portfolios. The final return for each group is then calculated as the average return across these 10 sub-portfolios. This approach helps minimize the impact of start-date-related anomalies, ensuring a more robust and unbiased performance evaluation.

Among the non-graph-based models, MLP shows the best performance with a positive excess annualized return(AR) of 5.23% on the CSI 1000. However, its performance drops significantly across other indices, yielding a negative AR of −3.09% on the CSI 300. LSTM and GRU generally perform poorly, with GRU generating negative returns across all indices except for a modest 0.4% AR on the CSI 300. These results suggest that traditional non-graph-based methods may not capture the nuanced, interdependent relationships in stock data, especially in portfolios like CSI 500.

The graph-based methods display stronger overall performance and better consistency across indices. GCN performs moderately well, especially in the CSI 1000, achieving a notable 4.25% AR. However, it shows lower returns in other indices, with a minimal negative AR in the CSI All Share (−1.32%) and CSI 300 (−4.60%), suggesting its effectiveness is largely concentrated on smaller-cap stocks. GLCN achieves high AR in the CSI 1000 (3.55%) but fails to generalize well across other indices. The poor generalization indicates that GLCN may overfit specific index characteristics rather than broadly applicable inter-stock relationships. HAD-GNN demonstrates strong, consistent performance, with a 9.39% AR in the CSI 1000 and a positive 1.77% AR in the CSI All Share. The method's multi-level aggregation and hierarchical modeling contribute to its efficacy, particularly in small-cap and broad indices. L-GCN performs relatively well, achieving a positive 6.17% AR in the CSI 1000 and moderate results on the CSI All Share (1.12%) and CSI 300 (0.79%). Its performance on CSI 1000 indicates its ability to model inter-stock relationships effectively in smaller-cap stocks, but its performance weakens on larger-cap indices.

IDGL shows poor performance across indices, with negative AR on the CSI All Share (−2.33%) and CSI 300 (−2.47%). It performs best on the CSI 1000 (2.84%), yet this result is modest compared to other graph-based methods, suggesting that IDGL's implicit graph construction method may not be robust for capturing complex financial interdependencies.

Table 5
Backtest results. The backtest strategy selects the top 10% of stocks with the highest predicted return ratio, and reports the resulting excess annualized return (AR).

Methods	CSI All Share	CSI 1000	CSI500	CSI300
MLP	−0.06%	5.23%	−0.37%	−3.09%
LSTM	0.44%	0.30%	−2.12%	0.86%
GRU	−2.13%	−0.46%	−0.66%	0.40%
GCN	−1.32%	4.25%	−0.20%	−4.60%
L-GCN	1.12%	6.17%	0.18%	0.79%
IDGL	−2.33%	2.84%	−0.25%	−2.47%
AGCN	−0.41%	6.87%	−1.67%	1.01%
GLCN	0.71%	3.55%	−0.73%	0.63%
HAD-GNN	1.77%	9.39%	1.76%	0.09%
HSGNN(customer)	6.34%	9.49%	1.96%	2.27%
HSGNN(supplier)	**7.54%**	**12.14%**	**5.22%**	**3.67%**

Graph-based methods, particularly HSGNN, outperform non-graph-based approaches, especially in capturing complex stock interdependencies in indices like the CSI 1000. HSGNN excels due to its unique combination of explicit and implicit graph structures, with the explicit structure boosting its predictive power across multiple indices.

HSGNN(supplier) achieves strong performance across all indices, with an impressive AR of 12.14% on the CSI 1000 and positive returns across all CSI indices, including 7.54% on the CSI All Share, 5.22% on the CSI 500 and 3.67% on the CSI 300. HSGNN (customer) still performs competitively, achieving an AR of 6.34%, only slightly below that of the supplier-based HSGNN model. This performance underscores HSGNN's ability to effectively leverage both explicit and implicit relationships, showcasing its robustness across diverse stock portfolios. The supplier-based explicit structure, combined with the money-flow-based implicit structure, gives HSGNN a distinct advantage over methods that lack structured relational data.

6.5. Ablation study

To validate the design choices in our proposed framework, we perform an ablation experiment by removing three components individually: edge weight(w/o edge), implicit graph learning(w/o implicit), and explicit graph(w/o explicit). The experiment is performed on the all A-share dataset, and the results are presented in Table 6.

After removing edge weights of money-flow graphs, the model's IC and ICIR decreased to 12.40% and 1.25, respectively, with RankIC dropping to 10.49%. Although the performance reduction is relatively minor, it indicates that edge weights play an important role in fine-tuning the relationship strengths between nodes. Retaining edge weights enables HSGNN to achieve higher predictive accuracy by refining the multi-level relationships between stocks, highlighting the value of edge weights in capturing the nuances of stock dependencies.

When the implicit graph learning module is removed, IC, ICIR, and RankIC are 12.32%, 1.20, and 10.58%, respectively.

This result underlines the importance of implicit graph learning as a supplement to explicit structural information. Implicit graph learning captures latent, non-linear stock relationships, adding a layer of adaptability to the model beyond fixed supply chain connections.

Removing the explicit graph component caused IC, ICIR, and RankIC to fall to 12.47%, 1.21, and 10.84%, respectively. This had a particularly notable impact, indicating that explicit graph information is crucial for conveying structured relational information. The explicit graph component provides the model with defined supply chain relationships, which significantly aid in capturing stock interdependencies. Without this structured information, the model's capability to leverage known economic connections declines, leading to reduced overall performance. In terms of AR, HSGNN (customer) and HSGNN (supplier) achieve the highest returns across all CSI indices, demonstrating the efficacy of combining explicit and implicit graphs.

In conclusion, each component in HSGNN plays a crucial role, with their combination leading to the model's enhanced overall performance. The implicit graph is particularly valuable in capturing latent, non-linear stock relationships beyond predefined connections, while explicit graph learning and edge weights respectively contribute supplementary information on hidden dependencies and modulate relationship strengths. The full model outperforms all ablated versions, underscoring the complementary nature of these modules and their interdependence. This modular combination empowers HSGNN to effectively model complex, multi-layered stock relationships, which is essential in financial applications.

7. Discussions

7.1. Theoretical contributions

Our study advances the theoretical foundation of financial graph representation learning by addressing key limitations in existing methods. Unlike prior work that treats explicit and implicit graphs separately, we propose a dynamic framework that integrates domain knowledge with data-driven dependencies. By incorporating edge weights from explicit graphs into the attention-based learning process, we improve both the robustness and the interpretability of the learned graph structure. Specifically, the guidance graphs used in our framework are temporal money-flow explicit graphs, where edge weights capture correlated investor behaviors and thereby overcome the static nature of conventional financial networks. To ensure economic plausibility, we introduce a sign alignment principle that enforces direction consistency between learned dependencies and observed relationships, mitigating spurious correlations, and preserving global structure. Furthermore, we propose a dual-path message passing mechanism that explicitly separates and cross-propagates positive and negative edge influences. This design enables the model to learn cooperative and competitive firm interactions separately, enhancing the expressiveness of node representations. By modeling these relationships through a dual-channel architecture, our approach more accurately captures real-world dynamics. Collectively, these contributions offer a principled and realistic approach to financial graph learning, bridging the gap between explicit economic structures and latent market signals. Our framework enhances theoretical understanding and provides a solid foundation for improved predictive performance and systemic insight in financial applications.

7.2. Practical contributions

Our proposed framework offers several real-world benefits for investors, asset managers, and policymakers. By capturing both long-term structural connections such as supplier–customer relationships and short-term co-movement reflected in simultaneous changes in money-flow, the framework supports more informed and responsive

Table 6
The effect of components. Ablation experiments are conducted by individually removing three components: edge weight (w/o edge), implicit graph learning (w/o implicit), and explicit graph (w/o explicit).

	CSI all share index				CSI 1000				CSI 500				CSI 300			
Methods	IC	ICIR	RankIC	AR	IC	ICIR	RankIC	AR	IC	ICIR	RankIC	AR	IC	ICIR	RankIC	AR
w/o edge	12.40%	1.25	10.49%	1.52%	10.96%	1.03	9.77%	8.28%	8.28%	0.65	7.08%	2.27%	4.89%	0.33	4.13%	−3.51%
w/o implicit	12.32%	1.20	10.58%	0.30%	11.13%	1.05	10.05%	7.41%	8.52%	0.67	7.25%	−0.07%	4.77%	0.33	4.13%	−3.77%
w/o explicit	12.47%	1.21	10.84%	4.20%	11.28%	1.05	10.33%	9.05%	8.65%	0.67	7.55%	2.06%	5.00%	0.35	4.16%	−0.83%
HSGNN (customer)	13.71%	1.38	11.97%	6.34%	12.34%	1.18	11.23%	9.49%	10.00%	0.80	8.84%	1.96%	7.02%	0.49	6.34%	2.27%
HSGNN (supplier)	**14.31%**	**1.45**	**12.43%**	**7.54%**	**12.81%**	**1.25**	**11.67%**	**12.14%**	**10.40%**	**0.84**	**9.11%**	**5.22%**	**7.25%**	0.52	6.39%	3.67%

investment decisions. The structure-aware implicit graph, driven by real-time transaction data and BARRA risk factors, uncovers latent yet influential relationships such as co-exposure to macroeconomic shocks or behavioral contagion effects that are often invisible in static financial networks. For example, during periods of market stress, the implicit graph can discover stocks that exhibit synchronized declines when interest rates rise sharply, particularly in rate-sensitive sectors such as utilities, even if they lack direct supply-chain connections. This allows portfolio managers to identify hidden sources of systemic risk and implement more robust hedging strategies. In addition, regulators can use similar graph-based diagnostics to monitor the emergence of fragile market clusters or identify firms disproportionately affected by collective market behavior.

In summary, our work not only advances the theory of graph learning in temporal financial systems but also delivers actionable insights that can improve risk management, portfolio construction, and market surveillance.

7.3. Limitations and future studies

In the future, we plan to explore a broader range of dynamic stock correlations and evaluate their effectiveness across different markets. In addition, we aim to incorporate alternative data sources such as financial news, analyst reports, and social media sentiment into our predictive framework to further enhance decision-making performance.

Nevertheless, our current work presents several limitations that future research could address. First, while we primarily focus on structured, domain-specific data such as supply chain relationships and money flow signals, this limits the model's access to unstructured, event-driven market information. A promising direction is to incorporate alternative textual signals using large language models, which can extract sentiment polarity from financial news and help capture investor expectations beyond historical transactions. Second, although our framework integrates explicit supply chain graphs and structure-aware implicit graphs, it does not fully explore interactions across different data modalities. Future work could develop a multi-source fusion architecture that aligns textual events with structured graphs through knowledge-aware attention and dynamic gating mechanisms, enabling a richer and more contextual understanding of firm dynamics.

8. Conclusion

In stock markets, the price momentum of one stock often influences the returns of other interconnected stocks, a phenomenon known as the momentum spillover effect. Despite its importance, relatively few studies have leveraged the dynamic interactions among firms for trend prediction. To address this gap, we propose HSGNN, a novel graph-based framework that captures shared information among stocks by jointly modeling explicit and implicit firm relationships, thereby significantly enhancing stock return prediction.

Specifically, we construct dynamic, weighted money-flow explicit graphs as guidance graphs, where edge weights capture correlated investor behavior and mitigate the limitations of static financial networks. This information is integrated into an implicit graph learning process to better reflect evolving market dynamics. To ensure economic plausibility and reduce spurious correlations, we introduce a sign consistency constraint that aligns the direction of learned dependencies with real-world relationships. Moreover, a dual-path message passing mechanism separately models cooperative and competitive interactions, while a cross-graph attention module incorporates slow-changing explicit supply chain structures to capture complex, multifaceted firm relationships. These designs collectively enhance the expressiveness of node representations and improve predictive performance.

We demonstrate the effectiveness and robustness of our proposed framework through extensive experiments on the Chinese market. Experiments on the six-year data of the CSI300, CSI 500, CSI1000 demonstrate the superiority of the proposed framework over state-of-the-art algorithms, including MLP-based methods, time series methods, and various graph-based approaches with enhancements in terms of the IC, ICIR and RankIC, respectively. Furthermore, the backtest results demonstrate that our solution enhances investors' decision-making capabilities, as the portfolio consistently outperforms the benchmark across different stock pools, delivering positive excess returns.

CRediT authorship contribution statement

Yang Du: Writing – review & editing, Writing – original draft, Methodology. **Biao Li:** Writing – review & editing, Validation, Investigation. **Zhichen Lu:** Validation, Project administration, Data curation. **Gang Kou:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Project administration, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is partially supported by grants from the National Natural Science Foundation of China (#W2511077 and #72495125), funding from Xiangjiang Laboratory (#25XJ02002), and the science and technology innovation Program of Hunan Province #2024RC4008.

Data availability

The authors do not have permission to share data.

References

[1] C.-F. Tsai, Y.-C. Hsiao, Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches, Decis. Support Syst. 50 (1) (2010) 258–269.
[2] E.F. Fama, K.R. French, Common risk factors in the returns on stocks and bonds, J. Financ. Econ. 33 (1) (1993) 3–56.
[3] M.M. Carhart, On persistence in mutual fund performance, J. Financ. 52 (1) (1997) 57–82.

[4] T. Yang, T.R. Yu, H. Zhao, Uncovering the relationship between incidental emotion toward a disaster and stock market fluctuations: Evidence from the US market, Decis. Support Syst. 181 (2024) 114213.
[5] S. Giglio, B. Kelly, D. Xiu, Factor models, machine learning, and asset pricing, Annu. Rev. Financ. Econ. 14 (2022).
[6] H.C. Schmitz, B. Lutz, D. Wolff, D. Neumann, When machines trade on corporate disclosures: using text analytics for investment strategies, Decis. Support Syst. 165 (2023) 113892.
[7] Z. Shao, X. Yao, F. Chen, Z. Wang, J. Gao, Revisiting time-varying dynamics in stock market forecasting: A multi-source sentiment analysis approach with large language model, Decis. Support Syst. (2024) 114362.
[8] D. Aobdia, J. Caskey, N.B. Ozel, Inter-industry network structure and the cross-predictability of earnings and stock returns, Rev. Account. Stud. 19 (2014) 1191–1224.
[9] Y. Shi, Y. Qu, Z. Chen, Y. Mi, Y. Wang, Improved credit risk prediction based on an integrated graph representation learning approach with graph transformation, European J. Oper. Res. 315 (2) (2024) 786–801.
[10] Y. Chen, Z. Wei, X. Huang, Incorporating corporation relationship via graph convolutional neural networks for stock price prediction, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1655–1658.
[11] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, T.-S. Chua, Temporal relational ranking for stock prediction, ACM Trans. Inf. Syst. (TOIS) 37 (2) (2019) 1–30.
[12] R. Cheng, Q. Li, Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, (1) 2021, pp. 55–62.
[13] C. Zhong, W. Du, W. Xu, Q. Huang, Y. Zhao, M. Wang, LSTM-ReGAT: A network-centric approach for cryptocurrency price trend prediction, Decis. Support Syst. 169 (2023) 113955.
[14] T.J. Moskowitz, M. Grinblatt, Do industries explain momentum? J. Financ. 54 (4) (1999) 1249–1290.
[15] C.A. Parsons, R. Sabbatucci, S. Titman, Geographic lead-lag effects, Rev. Financ. Stud. 33 (10) (2020) 4721–4770.
[16] L. Cohen, A. Frazzini, Economic links and predictable returns, J. Financ. 63 (4) (2008) 1977–2011.
[17] L. Menzly, O. Ozbas, Market segmentation and cross-predictability of returns, J. Financ. 65 (4) (2010) 1555–1580.
[18] C. Li, R. Li, X. Diao, C. Wu, Market segmentation and supply-chain predictability: evidence from China, Account. Financ. 60 (2) (2020) 1531–1562.
[19] Y. Ye, S. Ji, Sparse graph attention networks, IEEE Trans. Knowl. Data Eng. 35 (1) (2021) 905–916.
[20] N. Barberis, A. Shleifer, J. Wurgler, Comovement, J. Financ. Econ. 75 (2) (2005) 283–317.
[21] S. Titman, C. Wei, B. Zhao, Corporate actions and the manipulation of retail investors in China: An analysis of stock splits, J. Financ. Econ. 145 (3) (2022) 762–787.
[22] T. Zhang, Y. Li, Y. Jin, J. Li, AutoAlpha: an efficient hierarchical evolutionary algorithm for mining alpha factors in quantitative investment, 2020, arXiv preprint arXiv:2002.08245.
[23] C. Cui, W. Wang, M. Zhang, G. Chen, Z. Luo, B.C. Ooi, Alphaevolve: A learning framework to discover novel alphas in quantitative investment, in: Proceedings of the 2021 International Conference on Management of Data, 2021, pp. 2208–2216.
[24] S. Yu, H. Xue, X. Ao, F. Pan, J. He, D. Tu, Q. He, Generating synergistic formulaic alpha collections via reinforcement learning, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5476–5486.
[25] Z. Kakushadze, 101 formulaic alphas, Wilmott 2016 (84) (2016) 72–81.
[26] X. Yang, W. Liu, D. Zhou, J. Bian, T.-Y. Liu, Qlib: An ai-oriented quantitative investment platform, 2020, arXiv preprint arXiv:2009.11189.
[27] L. Cohen, D. Lou, Complicated firms, J. Financ. Econ. 104 (2) (2012) 383–400.
[28] C.M. Lee, S.T. Sun, R. Wang, R. Zhang, Technological links and predictable returns, J. Financ. Econ. 132 (3) (2019) 76–96.
[29] W. Xu, W. Liu, C. Xu, J. Bian, J. Yin, T.-Y. Liu, Rest: Relational event-driven stock trend forecasting, in: Proceedings of the Web Conference 2021, 2021, pp. 1–10.
[30] J. Gao, X. Ying, C. Xu, J. Wang, S. Zhang, Z. Li, Graph-based stock recommendation by time-aware relational attention network, ACM Trans. Knowl. Discov. from Data (TKDD) 16 (1) (2021) 1–21.
[31] Z. Zheng, J. Shao, J. Zhu, H.T. Shen, Relational temporal graph convolutional networks for ranking-based stock prediction, in: 2023 IEEE 39th International Conference on Data Engineering, ICDE, IEEE, 2023, pp. 123–136.
[32] H. Tian, X. Zheng, K. Zhao, M.W. Liu, D.D. Zeng, Inductive representation learning on dynamic stock co-movement graphs for stock predictions, INFORMS J. Comput. 34 (4) (2022) 1940–1957.
[33] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, Q. Su, Modeling the stock relation with graph network for overnight stock movement prediction, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4541–4547.
[34] G. Song, T. Zhao, S. Wang, H. Wang, X. Li, Stock ranking prediction using a graph aggregation network based on stock price and stock relationship information, Inform. Sci. 643 (2023) 119236.
[35] Msci Inc., The barra China equity model (CNE5), 2017, URL https://www.msci.com/www/research-paper/the-barra-china-equity-model/014459336. Accessed 03 August 2024.
[36] L.C.M. Phuong, Investor sentiment by money flow index and stock return, Int. J. Financ. Res. 12 (4) (2021) 33–42.
[37] Y. Chen, L. Wu, M.J. Zaki, Graphflow: Exploiting conversation flow with graph neural networks for conversational machine comprehension, 2019, arXiv preprint arXiv:1908.00059.
[38] B. Gao, C. Yang, Investor trading behavior and sentiment in futures markets, Emerg. Mark. Financ. Trade 54 (3) (2018) 707–720.
[39] F. Meng, C. Chen, Y. Ye, W.-B. Huang, Stock price co-movement prediction based on stock market technique indicators, Procedia Comput. Sci. 242 (2024) 1058–1065.
[40] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[41] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.
[42] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[43] D. Huang, P. Chen, R. Zeng, Q. Du, M. Tan, C. Gan, Location-aware graph convolutional networks for video question answering, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, (07) 2020, pp. 11021–11028.
[44] Y. Chen, L. Wu, M. Zaki, Iterative deep graph learning for graph neural networks: Better and robust node embeddings, Adv. Neural Inf. Process. Syst. 33 (2020) 19314–19326.
[45] R. Li, S. Wang, F. Zhu, J. Huang, Adaptive graph convolutional neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, (1) 2018.
[46] B. Jiang, Z. Zhang, D. Lin, J. Tang, B. Luo, Semi-supervised learning with graph learning-convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11313–11320.
[47] Z. Ding, R.D. Martin, The fundamental law of active management: Redux, J. Empir. Financ. 43 (2017) 91–114.
[48] T.T. Huynh, M.H. Nguyen, T.T. Nguyen, P.L. Nguyen, M. Weidlich, Q.V.H. Nguyen, K. Aberer, Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 850–858.
[49] L. Zhao, S. Kong, Y. Shen, Doubleadapt: A meta-learning approach to incremental learning for stock trend forecasting, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 3492–3503.

Yang Du is currently a Ph.D. candidate in School of Business Administration, Southwestern University of Finance and Economics. Her research interests include asset pricing and machine learning, graph deep learning.

Biao Li is an associate professor at the research institute of big data, Southwestern University of Finance and Economics. His research areas include multi-modal models, computer vision models, and AI finance models. He received his Ph.D. in Management Science and Engineering from the University of Chinese Academy of Sciences and master degree in Business and Administration from Tianjin University. He have published several academic papers in SCI journals and presented at international data mining conferences such as ICDM and ITQM, as well as at domestic CCF BIGDATA conferences.

Zhichen Lu is currently a Quantitative Researcher at Wanjia Mutual Fund and an external researcher at Southwestern University of Finance and Economics. His research focuses on deep learning, graph learning, and their applications in quantitative investment. He has experience in quantitative research, investment, and trading management at Huatai Securities Co., Ltd. and CSC Financial Co., Ltd. He received his master degree from the University of Chinese Academy of Sciences. His academic research in these areas has been cited hundreds of times.

Gang Kou is a Distinguished Professor of Chang Jiang Scholars Program at Xiangjiang Laboratory and Southwestern University of Finance and Economics, managing editor of International Journal of Information Technology & Decision Making (SCI) and managing editor-in-chief of Financial Innovation (SSCI). He is also editors for the following journals: European Journal of Operational Research, and **Decision Support Systems**. Previously, he was a professor of School of Management and Economics, University of Electronic Science and Technology of China, and a research scientist in Thomson Co., R&D. He received his Ph.D. in Information Technology from the College of Information Science & Technology, Univ. of Nebraska at Omaha; Master degree in Dept of Computer Science, Univ. of Nebraska at Omaha; and B.S. degree in Department of Physics, Tsinghua University, China. He has published more than 100 papers in various peer-reviewed journals.