Zero-Shot Extraction of Stock Relationship Graphs with LLMs

Hao Zhou[1] **Luis Felipe Costa Sperb**[1,2] **Tiejun Ma**[1]
[1]School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom
[2]Kingston Business School, Kingston University, London, United Kingdom
{hao.zhou, v1lcost5, tiejun.ma}@ed.ac.uk
f.costasperb@kingston.ac.uk

Abstract

Stock return prediction using Graph Neural Networks (GNNs) is often hindered by flawed graph structures. Existing models typically rely on rigid, predefined static graphs based on industry classifications or knowledge bases, which fail to capture the nuanced and complex business relationships between companies. To address this limitation, we pioneer the use of Large Language Models (LLMs) for zero-shot extraction of stock relationship graphs. By prompting an LLM, we extract its prior knowledge to construct a multi-relational static graph that captures fundamental corporate relationships. This method eliminates the reliance on simplistic, predefined industry classifications or knowledge base. To our knowledge, this is the first work to leverage zero-shot LLM graph generation for financial modeling, providing a more meaningful structural backbone for GNN-based prediction tasks.

1 Introduction

Stock return prediction is a crucial technique for profitable stock investment, and recent studies have begun to incorporate stock relationships as additional information for forecasting. To explore such information, graph neural networks (GNNs), a powerful paradigm for modeling inter-stock dependencies, are being applied. However, the predictive power of GNNs is often constrained by inadequate graph construction strategies. Current approaches that use static graphs rely on predefined relationships (e.g., industry sectors) that cannot capture evolving business relationships. In reality, stocks are not independent and can be influenced by complex connections beyond simple sector groupings; for instance, competitive or supply-chain relationships create dependencies that predefined classifications miss. These rigid graphs fail to capture meaningful underlying relationships, limiting the GNN's potential.

To address this significant gap in graph construction, we introduce a novel method for creating a static relationship graph to serve as the market's "structural backbone". By prompting a Large Language Model (LLM), we extract fundamental business relationships that reflect stable, long-term interconnections between companies, including but not limited to sector connections, competitive relationships, and supply chain dependencies. Our approach moves beyond the traditional, structurally-defined graphs used in prior research.

To sum up, our core contribution is that we pioneer the use of LLMs for zero-shot extraction of stock relationship graphs that capture multifaceted business relationships and long-term structural interconnections between companies, eliminating reliance on predefined industry classifications. To the best of our knowledge, this is the first work to prompt LLMs for this purpose in the financial domain.

2 Related Work

Graph Neural Networks in Finance Patel et al. (2024) identified the common pattern and segregated this task into three different modules: Graph Construction Module, Historical Information Encoder and Relational Module. Early studies typically rely on predefined stock relationships, such as industry-sector (Sawhney et al., 2021), consumer-supplier (Chen and Robert, 2022), and shareholding patterns (Wang et al., 2023), etc. Some works also construct static correlation graphs based on historical stock price (Li et al., 2021; Yin et al., 2021), though they are more widely used in build dynamic graphs due to their fast-changing nature. For instance, Cheng and Li (2021) infer the latent stock relation from the sequential embedding at each timestep. Since the static graphs and dynamic graphs model the stock relationships from different views, researchers has began to explore the

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

combination of them. For example, Wang et al. (2022) use a static graph which is predefined based on domain knowledge and a latent dynamic graph which is learned end-to-end. The output feature vectors from the two separate graph convolutions are summed together to create a single fused representation.

LLMs in Finance Recent researches begin to explore the potential of using information extracted by LLMs to construct and analyze knowledge graphs in the financial sector. Notably, Trajanoska et al. (2023) used LLMs to generate knowledge graphs from ESG (Environmental, Social, and Governance) reports, creating node-edge-node triples to represent relationships between entities, including companies. Similarly, Cheng et al. (2022) developed a Semantic-Entity Interaction Module with LLMs and CRF to construct financial knowledge graphs from brokerage reports, demonstrating the potential of zero-shot techniques for relationship extraction without manual rule-setting. However, these works all rely on additional external textual information, while we aim to extract the prior knowledge within the LLMs to build company relationship graphs.

3 Methodology

3.1 Framework Overview

In our framework, we first employ an LLM to perform zero-shot extraction of structured company relationship graphs without any textual input or external data sources, thereby allowing us to directly probe the LLM's prior knowledge about inter-company relationships. The extracted graph encodes multiple types of relations, including supply chain dependencies, competitive dynamics, and strategic partnerships, represented as multi-relational edges. The initial node features are constructed from historical stock price data. To model the structural and semantic information embedded in these graphs, we adopt two representative graph neural network (GNN) architectures: the Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2017) and the Relational Graph Attention Network (RGAT) (Busbridge et al., 2019). The learned node representations are subsequently used for stock ranking prediction. In summary, the framework enables us to evaluate the efficacy of the LLM-extracted relationship graphs by comparing their predictive performance against models trained on other predefined relationship graphs.

3.2 LLM-prompted Static Graph Construction Module

The foundation of our model is a static, multi-relational graph, G_S, designed to capture the multifaceted, long-term economic ties between companies. These fundamental relationships, such as supply chains and competitive positions, provide a structural backbone that is less susceptible to the daily noise of market news. Instead of relying on manually curated databases, which can be incomplete or outdated, we introduce a novel methodology to construct this graph by leveraging a Large Language Model (LLM) as a zero-shot knowledge extractor.

The first step is to systematically query the LLM to identify relationships between every pair of companies (s_i, s_j) in our stock universe S. To ensure the LLM provides structured and relevant output, we employ carefully designed prompt engineering.

Based on the S&P Global Business Relationship Dataset [1], we define a comprehensive **multi-relational taxonomy**, $\mathcal{R}$, that covers key economic interactions: is_Customer_of, is_Supplier_of, is_Distributor_of, is_Competitor_of, is_Peer_of, is_Investor_of, is_Invested_by, is_Subsidiary_of, is_Parent_of, is_Cross_owned_with, is_Joint-Venture_partner_of, is_R&D_partner_of, is_Marketing_partner_of, is_Strategic_partner_of, is_Licensor_of, is_Licensee_of, is_Franchisor_of, is_Franchisee_of, is_Creditor_of, is_Borrower_of, is_Acquirer _of, is_Target_of_acquisition, is_Merger_partner_with, has_Interlocking_directors_with.

For each pair of companies, we use a structured prompt that forces the LLM to classify their primary relationship into one of these predefined categories and to provide a confidence score. An example prompt is shown in Figure 1.

However, LLM-generated graphs can be sparse or noisy for certain relation types. To ensure that each relational graph used by our model possesses a meaningful level of connectivity, we apply a **connectivity-based pruning** step. Specifically, for each relation type r, we calculate the total number of edges in its initial graph, given by $\|A'_r\|_1 = \sum_{i,j} A'_r[i,j]$. If this edge count falls below a predefined connectivity threshold κ (set to

[1]https://www.marketplace.spglobal.com/en/datasets/business-relationships-(5)

```
You are a financial analyst with expertise
in the US market. For each of the following
relationships, please help me find five
companies from the list of SP500
constituents which have that relationship
with the source company and sort them from
high to low by relevance. It's fine if you
can't find enough related companies. Please
make sure the relationship is existing and
real. The companies are represented by
ticker symbol.

Your response should be in the json format
without explanation: {{{Source_company}:
{{Relation_1: [company_1,
company_2,...]}}}}.

Source company: {ticker}
Relationships: {Relation_1, Relation_2,...}
```

Figure 1: The structured prompt used to query the LLM for zero-shot relationship extraction between company pairs.

200 in our experiments), we deem the relation type too sparse to be reliable and discard it entirely.

This pruning step yields a final, refined set of adjacency matrices $\{A_r\}_{r\in\mathcal{R}_{final}}$, where $\mathcal{R}_{final} \subseteq \mathcal{R}$. Collectively, the set of nodes S and these filtered adjacency matrices constitute our static multi-relational graph, $G_S = (S, \{A_r\}_{r\in\mathcal{R}_{final}})$, which serves as a stable and robust input to the dual-component GNN encoder.

3.3 Graph Neural Networks

We employ two representative multi-relational graph neural networks to model the extracted company relationship graphs: the Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2017) and the Relational Graph Attention Network (RGAT) (Busbridge et al., 2019). RGCN extends standard graph convolutions to multi-relational settings by applying relation-specific transformations with parameter-efficient basis decomposition, enabling effective aggregation of neighborhood information across different relation types. RGAT, in contrast, incorporates relation-aware attention mechanisms that adaptively weight the influence of neighboring nodes, allowing the model to focus on more informative relations. Both architectures operate on initial node features derived from historical price data, and their learned representations are used for stock ranking prediction.

RGAT The Relational Graph Attention Network (RGAT) (Busbridge et al., 2019) generalizes the conventional graph attention mechanism (Veličković et al., 2018) to accommodate multi-relational graphs by introducing relation-specific transformations and attention computations. For each relation type r, a relation-specific linear transformation $\mathbf{W}^r$ is applied to the input node features, producing

$$\mathbf{h}_i^r = \mathbf{W}^r \mathbf{x}_i. \tag{1}$$

Multi-head attention coefficients are then computed as

$$e_{ij}^{(r,h)} = \mathrm{LeakyReLU}\left(\mathbf{a}^{(r,h)\top}\left[\mathbf{h}_i^r \,\|\, \mathbf{h}_j^r\right]\right), \tag{2}$$

where $\mathbf{a}^{(r,h)}$ denotes a learnable attention vector associated with relation r and attention head h, and $\|$ represents vector concatenation. These coefficients are normalized via the softmax function:

$$\alpha_{ij}^{(r,h)} = \frac{\exp\left(e_{ij}^{(r,h)}\right)}{\sum_{k\in\mathcal{N}(i)} \exp\left(e_{ik}^{(r,h)}\right)}. \tag{3}$$

The relation-specific aggregated representation is then obtained as

$$\mathbf{h}_i'^{(r)} = \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(r,h)} \mathbf{h}_j^r. \tag{4}$$

Finally, the outputs across all relations are combined through an aggregation function (e.g., sum, mean, max, or learned attention-based weighting) to yield the final node representation:

$$\mathbf{h}_i' = \mathrm{AGG}\left(\mathbf{h}_i'^{(r)}\right)_{r=1}^{R}. \tag{5}$$

RGCN The Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2017) extends the standard graph convolution operation to multi-relational graphs by incorporating relation-specific transformations and a basis decomposition scheme for parameter efficiency. For each node i and relation type r, the model applies degree-based normalization with

$$c_{i,r} = \frac{1}{|\mathcal{N}_i^r|}, \tag{6}$$

where $\mathcal{N}_i^r$ denotes the set of neighbors of node i connected via relation r. The relation-specific transformation matrices $\mathbf{W}^r$ are parameterized using a basis decomposition:

$$\mathbf{W}^r = \sum_{b=1}^{B} a_{rb} \mathbf{V}_b, \tag{7}$$

where $\{\mathbf{V}_b\}_{b=1}^{B}$ are shared basis matrices and a_{rb} are learned relation-specific coefficients. This reduces the number of parameters from $R \times d_{\text{in}} \times d_{\text{out}}$ to $B \times d_{\text{in}} \times d_{\text{out}} + R \times B$. The forward propagation rule combines self-loop and neighbor messages as

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)}\mathbf{h}_i^{(l)} + \sum_{r=1}^{R}\sum_{j\in\mathcal{N}_i^r} c_{i,r}\mathbf{W}_r^{(l)}\mathbf{h}_j^{(l)}\right), \tag{8}$$

where $\mathbf{W}_0^{(l)}$ handles self-connections and $\sigma(\cdot)$ is a non-linear activation function. This formulation enables efficient learning over knowledge graphs while preserving relation-specific inductive biases through the decomposed weight matrices.

3.4 Training Objective

Following previous works (Feng et al., 2019; Li et al., 2024), we formulate the next-day stock return prediction as a *learning to rank* problem. We optimize a hybrid objective combining regression and ranking terms:

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \phi \cdot \mathcal{L}_{\text{rank}}, \tag{9}$$

where ϕ is a hyperparameter controlling the contribution of ranking supervision. We set it to be 0.5 here. The regression loss is defined as:

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i^{t+1} - r_i^{t+1}\right)^2, \tag{10}$$

where N is the number of stocks, $\hat{y}_i^{t+1}$ is the predicted score, and r_i^{t+1} is the ground truth return.

The ranking loss is formulated as:

$$\mathcal{L}_{\text{rank}} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\max\Big(0, (\hat{y}_i^{t+1} - \hat{y}_j^{t+1}) \cdot (r_j^{t+1} - r_i^{t+1})\Big), \tag{11}$$

which penalizes cases where the predicted ordering contradicts the ground truth ordering. For implementation, we compute all pairwise differences in predictions and ground truth, multiply them element-wise, and apply a ReLU to retain only positive ranking violations.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate on S&P 500 constituents using daily OHLCV data enriched with technical indicators (MA, RSI, MACD). Our chronological split spans training (2012/07-2022/06), validation (2022/07-2023/09), and testing (2023/10-2024/12). We filter out the companies which do not have full trading records during this period, resulting in 452 tickers in total. Critically, our static graph uses GPT-4o-mini (knowledge cutoff: October 2023) [2], with testing beginning October 2023 - ensuring true out-of-sample evaluation where LLM-extracted relationships are tested on genuinely unseen future data. See Appendix A for detailed feature descriptions and data splits.

Evaluation Metrics We evaluate our model using three complementary metric categories: ranking quality using Mean Reciprocal Rank (MRR), prediction accuracy with Mean Squared Error (MSE), and trading performance with the cumulative Investment Return Ratio (IRR) and Sharpe Ratio (SR). Specifically, We construct two long-only strategies by selecting the top-1 and top-10 stocks based on predicted rankings. At the start of the test period, we invest one unit of capital in each strategy and compute their cumulative profit and Sharpe ratio over the evaluation horizon.

Baselines We compare the proposed method against two commonly used predefined stock relationship graphs: (1) Wikidata-based relationships extracted from a structured knowledge base, and (2) the GICS industry-sector classification reflecting economic sector groupings. These baselines enable evaluation of the efficacy of the LLM-extracted relationship graph relative to established graph constructions. We follow the same process as Feng et al. (2019), the details are in Appendix B.

4.2 Relation Extraction Analysis

Before evaluating the performance on downstream tasks, it is crucial to verify that the LLM-extracted company relationship graph is structurally richer and more diverse than existing alternatives. We quantitatively compare the number and distribution of relations against two commonly used predefined graphs (GICS industry-sector hierarchy and Wikidata corporate relations). As shown in Table 2, compared to the baselines, our LLM-extracted graph exhibits clear advantages in relational diversity, coverage, and structural realism. It contains 13 distinct relation types, far exceeding the 4 in both Wikidata and GICS, enabling richer semantic mod-

[2] https://platform.openai.com/docs/models/gpt-4o-mini

Relations	MSE ($\times 10^{-4}$) $\downarrow$	MRR $\uparrow$	IRR (1) $\uparrow$	SR (1) $\uparrow$	IRR (10) $\uparrow$	SR (10) $\uparrow$
RGAT						
Wikidata	3.269	0.016	0.177	0.408	0.206	0.901
GICS	5.311	0.016	0.273	0.615	0.237	1.060
Ours (LLM)	**3.196**	**0.035**	**0.421**	**0.835**	**0.262**	**1.185**
RGCN						
Wikidata	3.190	0.026	0.254	0.558	0.258	1.038
GICS	3.190	0.022	0.600	1.011	0.301	1.090
Ours (LLM)	**3.189**	**0.027**	**0.820**	**1.176**	**0.350**	**1.246**

Table 1: Performance comparison of RGCN and RGAT models on different relationship graphs. We report the average performance across 40 runs. "(1)" and "(10)" mean the top-1 and top-10 strategy, respectively.

Graph	R	$\lvert E\rvert$	Nodes	Sym.
Wikidata	4	4,723	358	✓
GICS	4	19,732	452	✓
Ours(LLM)	13	12,800	452	✗

Table 2: Basic statistics of the relationship graphs. R denotes the number of distinct relation types, $|E|$ denotes the total number of edges, "Nodes" denotes the number of unique companies covered, and "Sym." indicates whether the graph is symmetric.

eling. While its total number of edges (12,800) is lower than GICS (20,636), it is substantially denser than Wikidata (4,723), striking a balance between diversity and connectivity. In terms of coverage, it spans 452 companies, matching GICS and surpassing Wikidata's 358, ensuring applicability to the full stock universe. For both Wikidata and our LLM-generated relations, we filtered out relation types with fewer than 200 edges to maintain graph connectivity, ensuring that the comparison focuses on meaningful and well-connected relations. Importantly, our graph incorporates directed edges, capturing asymmetric corporate relationships (e.g., supplier–customer) that symmetric baselines cannot represent, thereby offering a more realistic and informative foundation for downstream stock ranking tasks.

4.3 Stock Ranking Performance

As shown in Table 1, the LLM-extracted relations consistently outperform Wikidata and GICS across both RGAT and RGCN backbones. In terms of prediction accuracy, our method achieves the lowest MSE in all cases, showing that LLM-extracted relations better capture stock dynamics. The gains are even clearer in ranking quality: MRR nearly doubles under RGAT (0.035 vs. 0.016) and remains the strongest under RGCN, indicating that our approach identifies top-performing stocks more effectively.

The improvements translate into substantial trading benefits. For both top-1 and top-10 strategies, our relations deliver markedly higher cumulative returns and Sharpe ratios, with the RGCN backbone achieving over 1.0 in top-1 Sharpe ratio (1.176), demonstrating strong risk-adjusted profitability. These consistent gains across two distinct architectures highlight that the advantage comes from the richer, directional relations themselves rather than a specific model choice, validating our claim that LLM-extracted structures bridge the gap between generic knowledge graphs and actionable financial insights.

5 Conclusions & Future Work

This work demonstrates that high-quality stock relationship graphs extracted from large language models can significantly enhance stock ranking accuracy and trading performance compared to widely used knowledge sources such as Wikidata and GICS. By leveraging LLMs' implicit knowledge of corporate relationships, we achieve 37% higher investment returns and improved Sharpe ratios across multiple evaluation strategies. To the best of our knowledge, we are the first to use zero-shot LLM extraction for financial graphs, opening a new research direction - using LLMs as knowledge bases for finance. Future work will explore temporal dynamics to capture evolving corporate relationships while maintaining stability, and investigate multi-modal integration combining LLM knowledge with alternative data sources and market sentiment.

References

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla. 2019. Relational graph attention networks. *Preprint*, arXiv:1904.05811.

Qinkai Chen and Christian-Yann Robert. 2022. Multivariate realized volatility forecasting with graph neural network. In *Proceedings of the Third ACM International Conference on AI in Finance*, ICAIF '22, page 156–164, New York, NY, USA. Association for Computing Machinery.

Rui Cheng and Qing Li. 2021. Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):55–62.

Zehua Cheng, Lianlong Wu, Thomas Lukasiewicz, Emanuel Sallinger, Georg Gottlob, Georg Gottlob, André Hernich, Clemens Kupke, Thomas Lukasiewicz, Marcelo Arenas, and 1 others. 2022. Democratizing financial knowledge graph construction by mining massive brokerage research reports. In *Proceedings of the Workshops of the {EDBT/ICDT} 2022 Joint Conference, Edinburgh, UK, March 29, 2022*, volume 846, pages 453–473. CEUR-WS. org.

Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Trans. Inf. Syst.*, 37(2).

Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. 2024. Master: market-guided stock transformer for stock price forecasting. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. 2021. Modeling the stock relation with graph network for overnight stock movement prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Manali Patel, Krupa Jariwala, and Chiranjoy Chattopadhyay. 2024. A systematic review on graph neural network-based methods for stock market forecasting. *ACM Comput. Surv.*, 57(2).

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. 2021. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):497–504.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *Preprint*, arXiv:1703.06103.

Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing knowledge graph construction using large language models. *Preprint*, arXiv:2305.04676.

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

Changhai Wang, Hui Liang, Bo Wang, Xiaoxu Cui, and Yuwei Xu. 2022. Mg-conv: A spatiotemporal multi-graph convolutional neural network for stock market index trend prediction. *Computers and Electrical Engineering*, 103:108285.

Heyuan Wang, Tengjiao Wang, Shun Li, and Shijie Guan. 2023. Hatr-i: Hierarchical adaptive temporal relational interaction for stock trend prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6988–7002.

Xingkun Yin, Da Yan, Abdullateef Almudaifer, Sibo Yan, and Yang Zhou. 2021. Forecasting stock prices using stock correlation graph: A graph convolutional network approach. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

A Dataset Details

For each S&P 500 constituent, we collect daily Open, High, Low, Close prices and Volume (OHLCV)[3]. We compute standard technical indicators including Moving Averages (MA), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD). These raw values are normalized to form time-series feature vectors $\mathbf{x}_i^t$ that initialize node representations in our model.

Table 3 details our chronological split, designed to simulate realistic live trading conditions:

Split	Period	Days
Train	2012/07 - 2022/06	2517
Validation	2022/07 - 2023/09	313
Test	2023/10 - 2024/12	315

Table 3: Chronological split of the dataset for training, validation, and testing.

The validation set is used for hyperparameter tuning and early stopping. By aligning our test period start with GPT-4o-mini's knowledge cutoff[4], we ensure the fundamental relationships extracted

[3] https://paperswithbacktest.com/datasets/stocks-daily-price

[4] https://platform.openai.com/docs/models/gpt-4o-mini

by the LLM are evaluated on their ability to generalize to future, unseen market data, providing a rigorous assessment of predictive power.

B Baseline Details

GICS The Global Industry Classification Standard, jointly developed by MSCI and Standard & Poor's, is a widely adopted taxonomy for categorizing companies into a four-tier hierarchical structure consisting of sectors, industry groups, industries, and sub-industries. This system facilitates consistent classification and comparison of firms based on their primary business activities, enabling investors and analysts to construct sector-based investment strategies and benchmark performance. In our experiments, we utilize GICS four-level relationships as one of the baseline graphs. The statistics of each level are shown in Table 4.

Level	$\lvert E \rvert$	**Categories**
Sector	10,841	11
Industry Group	5,323	25
Industry	2,347	67
Sub-Industry	1,221	119

Table 4: Statistics of GICS relationships at different levels.

Wikidata Wikidata is a collaboratively edited knowledge base that provides structured information across a wide range of domains, including corporate entities and their interconnections. In our work, we extract company–company relationships from Wikidata by meta-paths defined by Feng et al. (2019). To ensure the connectivity and robustness of the resulting graph, we filter out relation types whose total number of connections is below 200, retaining only sufficiently frequent relations for analysis.

Meta-path	$\lvert E \rvert$
Industry - Industry	1,944
Member of - Member of	1,220
Owned by - Owned by	1,040
Product or material produced -	519

Table 5: Statistics of Wikidata relationships at different meta-paths.