Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Network-incorporated Portfolio Optimization with Graph Machine Learning

Jiu Zhang

zhangjiu@zju.edu.cn

Fuyao University of Science and Technology

Long Xiong

Yunnan University

Tingting Chen

Zhejiang University of Finance and Economics

Yan Li

Zhejiang Gongshang University

Xiongfei Jiang

Ningbo University of Finance & Economics

Article

Keywords:

Posted Date: September 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10721676/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Network-incorporated Portfolio Optimization with Graph Machine Learning

Jiu Zhang[1,2,*], Long Xiong[3,*], Tingting Chen[4], Yan Li[5] and Xiongfei Jiang[6,*]

[1]School of Digital Economy and Management, Fuyao University of Science and Technology, 350109, Fuzhou, PR China

[2]Digital Governance Laboratory, Fuyao University of Science and Technology, 350109, Fuzhou, PR China

[3]School of Physics and Astronomy, Yunnan University, 650091, Kunming, PR China

[4]School of Finance, Zhejiang University of Finance and Economics, Hangzhou 310018, PR China

[5]School of Finance, Zhejiang Gongshang University, Hangzhou 310018, PR China

[6]College of Finance, Ningbo University of Finance & Economics, 315175, Ningbo, PR China

*Corresponding authors. zhangjiu@zju.edu.cn; longxiong@zju.edu.cn; jiangxiongfei@nbufe.edu.cn

Abstract

Financial markets exhibit complex interdependencies that conventional portfolio optimization approaches fail to capture. Here, we integrate dynamic stock correlation networks with graph machine learning to investigate how multiscale network information informs portfolio construction. We develop a hybrid architecture combining graph neural networks with Transformers to encode node-level features, local topological neighborhoods, and global market dependencies. Across the Chinese and American markets, each additional scale of network information improves risk-adjusted returns and sharpens allocations at the stock level, while the selected stocks remain spread across sectors. SHAP analysis identifies distinct topological features driving weight variation across assets, with features of higher distributional entropy exhibiting stronger attribution importance. Our results demonstrate that financial network topology provides interpretable signals for portfolio optimization, revealing how graph machine learning can exploit complex market dependencies that traditional frameworks overlook.

Introduction

Financial markets are complex dynamic systems in which stocks are interconnected through correlation structures that evolve over time, rather than behaving as independent assets. Identifying and exploiting these interdependencies is therefore a central challenge in portfolio optimization. Complex networks provide a general framework for characterizing such dependencies and their temporal evolution[1–3], with applications spanning biological[4–6], social[7–9], and financial systems[10–12]. A fundamental question in network science is how structure shapes function. In portfolio optimization, this question takes on a more specific form: how should an evolving network structure inform the allocation of capital?

Traditional portfolio optimization frameworks answer this question by assumption, treating asset dependencies through static covariance estimates[13], yet stocks are interconnected through economic, sectoral, and broad market interactions that generate dynamic cross correlations[14–17]. Network filtering methods, including minimum spanning trees (MST) and planar maximally filtered graphs (PMFG), extract informative structure from dense correlation matrices[18,19]. These representations have linked network topology to portfolio outcomes: stocks in minimum-risk portfolios

tend to occupy peripheral MST positions[16], whereas systemic fragility correlates with network geometry[10]. Such observations have motivated portfolio strategies based on clustering[20,21] and centrality measures[22,23].

Despite this progress, translating network structure into portfolio allocations remains methodologically constrained. Existing approaches typically use manually selected topological indicators to prescribe selection or weighting rules[20–23], focusing on specific structural aspects such as centrality or clustering. This reliance on predefined indicators may fail to capture high-dimensional, nonlinear interactions among multiple topological features. Moreover, financial networks contain information across multiple scales, from node-level features to local neighborhoods and global market dependencies, yet how these multiscale signals can be jointly integrated into portfolio weights through an end-to-end framework remains unclear.

Graph machine learning renders such integration tractable[24–27]. Graph neural networks (GNNs) learn representations through neighborhood message passing[24,25], whereas graph Transformers capture global dependencies across all nodes via self-attention[26,27]. The two mechanisms thus address precisely the local and global scales at which financial networks carry information. In equity markets, however, such architectures have so far been applied mainly to return forecasting and stock ranking[28–32], with portfolio weights assigned in a separate downstream step rather than learned directly under operational constraints. The contribution of network structure to the resulting allocation is also rarely quantified, leaving unclear which topological features actually matter for portfolio decisions.

Here, we formulate portfolio optimization as a network-informed machine learning problem on dynamic stock networks. Daily return correlations are filtered by PMFG into an evolving network, and each stock is represented by its node-level topological features. The GNN aggregates local neighborhood information, while the Transformer captures global dependencies through self-attention. These representations combine to automatically learn portfolio weights. Furthermore, we construct a hierarchy of model variants to isolate contributions from multiple scales of network structures. SHAP attribution analysis further quantifies how topological features contribute to allocations. We evaluate this framework in Chinese and American stock markets. Across both, richer network information improves risk-adjusted returns and produces distinct patterns of weight concentration. The methodological contribution is an interpretable mapping from dynamic network structure to portfolio decisions.

Table 1. Data partitioning for model training and evaluation. Training, validation, and test partitions for the Chinese and American stock markets used in this study. Samples whose holding periods crossed partition boundaries are excluded. The last column reports CSI 300 / S&P 500 in that order.

Partition	Start date	End date	Graph samples	Stock count
Train	2022-07-01	2024-06-30	479	
Validation	2024-07-01	2024-12-31	115	300 / 503
Test	2025-01-01	2025-12-31	233	

Results

Methodological framework

In this paper, we develop a graph machine-learning framework to incorporate network topology into portfolio optimization. The framework consists of two stages, as shown in Fig. 1. First, we construct

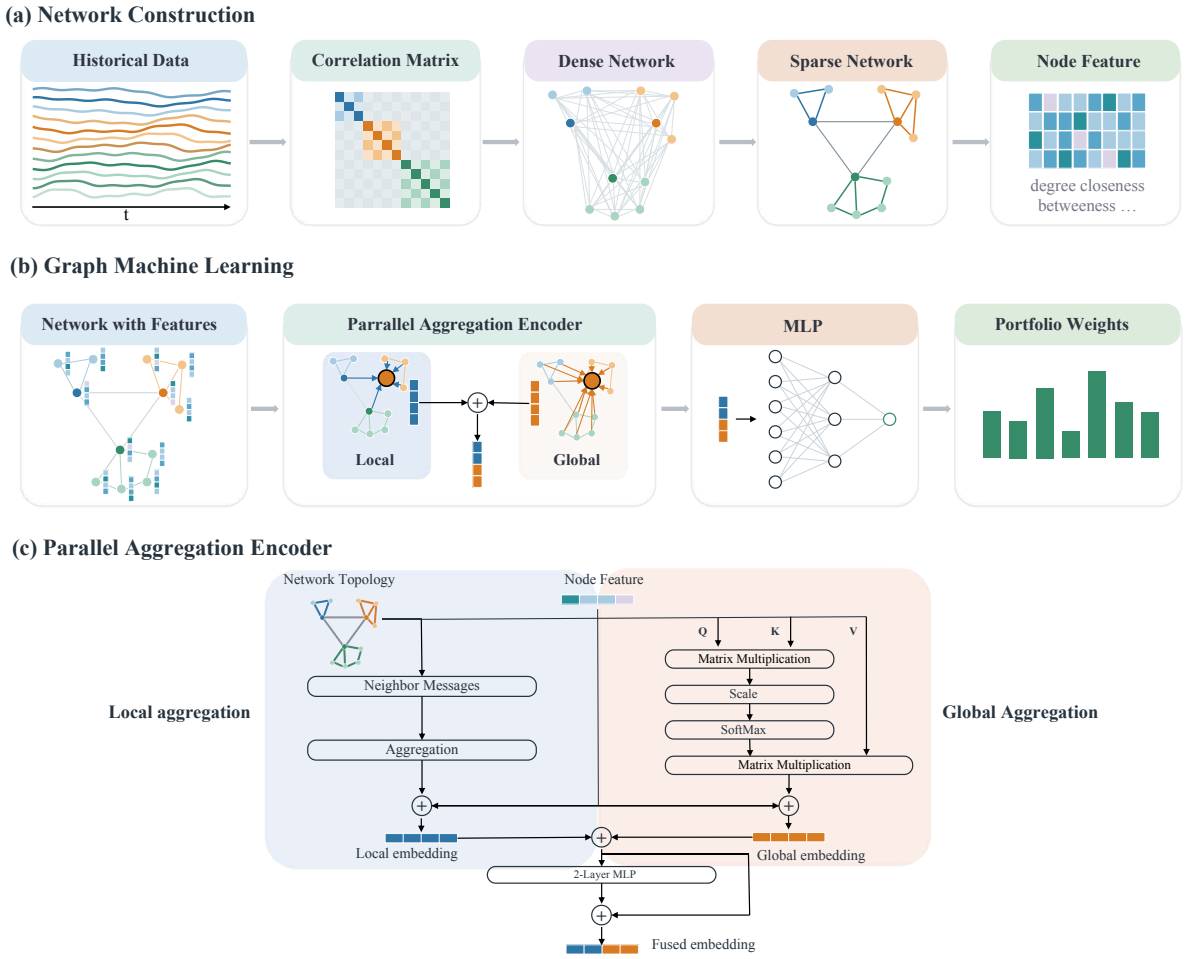


Fig. 1. Stock network construction and graph machine-learning framework for portfolio optimization. **a**, Network construction pipeline. Daily stock returns are transformed into correlation matrices, from which both dense and sparse networks are derived. The sparse network is filtered using the planar maximally filtered graph (PMFG) method. Thirty topological features are computed for each node, such as degree, clustering coefficient and betweenness centrality. (see Table 4). **b**, Graph machine-learning model architecture. The network adjacency matrix and node features are fed into a parallel aggregation encoder that captures local and global structural patterns simultaneously. A multilayer perceptron (MLP) maps the fused representation to portfolio weights. **c**, Parallel aggregation encoder design. The GNN module aggregates neighborhood information to learn local representations, whereas the Transformer module employs self-attention to capture global dependencies. Outputs from both modules are fused through a two-layer MLP.

dynamic networks from historical price returns. Second, we apply graph machine-learning models to learn from the network topology and node features for portfolio allocation.

In the first stage, we compute pairwise correlations over a rolling window of stock returns. On each trading day, we construct a network through the planar maximally filtered graph (PMFG) method to retain the most informative correlation links. For each stock, we compute 30 node-level topological features spanning six categories: degree-based, closeness-based, betweenness-based, and eigenvector-based centralities, plus community structure and structural-hole measures (Table 4). These node features and the network topology serve as inputs to the machine learning models (Fig. 1a).

Table 2. Portfolio performance metrics across markets and models. MDD denotes maximum drawdown. Bold values indicate the best performance among all strategies within each market.

Dataset	Strategy	Cumulative return	Sharpe	Sortino	MDD
CSI 300	Market	27.0%	1.76	1.96	**-11.0%**
	M	28.3%	1.85	1.99	-11.4%
	LM	44.6%	2.12	3.66	-13.8%
	GLM	**103.6%**	**2.65**	**4.08**	-13.9%
S&P500	Market	16.4%	0.98	1.23	-16.6%
	M	23.8%	1.41	1.75	-15.7%
	LM	26.7%	1.55	2.05	-15.3%
	GLM	**41.4%**	**2.12**	**2.77**	**-12.9%**

In the second stage, we design a hybrid graph machine learning model (GLM) that combines a GNN with a Transformer architecture. The GNN module captures local neighborhood information through message passing, whereas the Transformer module models global market-wide dependencies through self-attention (Fig. 1b,c). The outputs of the GNN and Transformer modules are fused through an MLP to produce portfolio allocation weights.

To investigate how network information at different scales contributes to portfolio optimization, we construct a hierarchy of model variants with progressively broader structural information. The M model uses only node-level topological features with an MLP to generate portfolio weights. The LM model extends M by incorporating local neighborhood information through the GNN module. The GLM model further extends LM by integrating global market dependencies through the Transformer module (Fig. 1c). This hierarchical design enables controlled evaluation of how different scales of network information influence portfolio allocation.

Portfolio performance

To evaluate the proposed graph machine-learning framework, we conduct experiments in the Chinese (CSI 300) and American (S&P 500) markets. Training, validation, and test partitions are summarized in Table 1.

A consistent performance hierarchy emerges across both markets (Fig. 2 and Table 2). In the Chinese market, the cumulative return increases from 28.3% for M to 44.6% for LM and 103.6% for GLM. The corresponding Sharpe ratio rises from 1.85 to 2.12 and 2.65. The market benchmark achieves a cumulative return of 27.0% with a Sharpe ratio of 1.76.

A similar pattern is observed in the American market, which exhibits the same ordering, where cumulative return increases from 23.8% for M to 26.7% for LM and 41.4% for GLM, accompanied by an increase in Sharpe ratio from 1.41 to 1.55 and 2.12. The market benchmark achieves a cumulative return of 16.4% with a Sharpe ratio of 0.98.

The LM model incorporating local neighborhood information yields higher returns and risk-adjusted performance relative to the M baseline. The GLM model adding global market dependencies provides larger performance gains. GLM achieves the highest cumulative return, and Sharpe ratio in both markets.

In the American market, GLM exhibits the smallest maximum drawdown among all evaluated portfolios, reaching -12.9% compared with -16.6% for the market benchmark. In the Chinese market, GLM's maximum drawdown is -13.9%, larger than the benchmark value of -11.0%. However, GLM's substantially higher return results in the highest Sharpe and Sortino ratios among

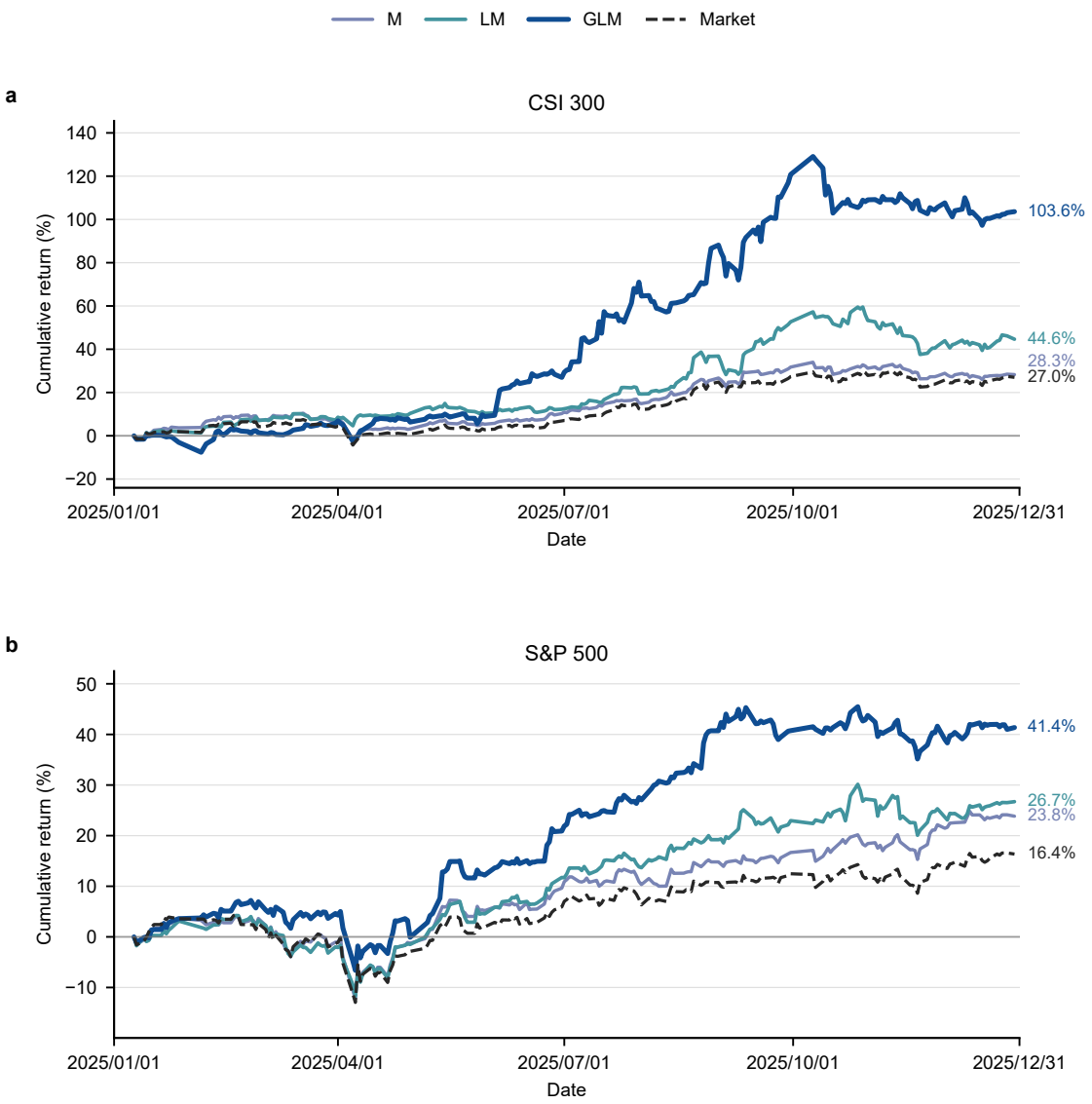


Fig. 2. Portfolio performance across network information scales. Cumulative returns from 1 January to 31 December 2025 for **a**, CSI 300 and **b**, S&P 500. M is a node-feature multilayer perceptron (MLP) baseline. LM extends M with local graph neural network (GNN) aggregation. GLM further integrates global Transformer self-attention with the GNN. Market represents an equal-weight portfolio of all constituent stocks.

all strategies.

Portfolio weight concentration

We examine how network information aggregation influences portfolio weight concentration. Portfolio weights evolve over time and exhibit distinct concentration patterns across the model hierarchy (Fig. 3a–f). The M model produces relatively diffuse allocations, with weights distributed across many stocks. The LM model generates more concentrated allocations with stronger preferences for selected stocks. The GLM model exhibits the highest selectivity, with substantial weight concentrated in a small number of stocks. This pattern is consistent across both markets.

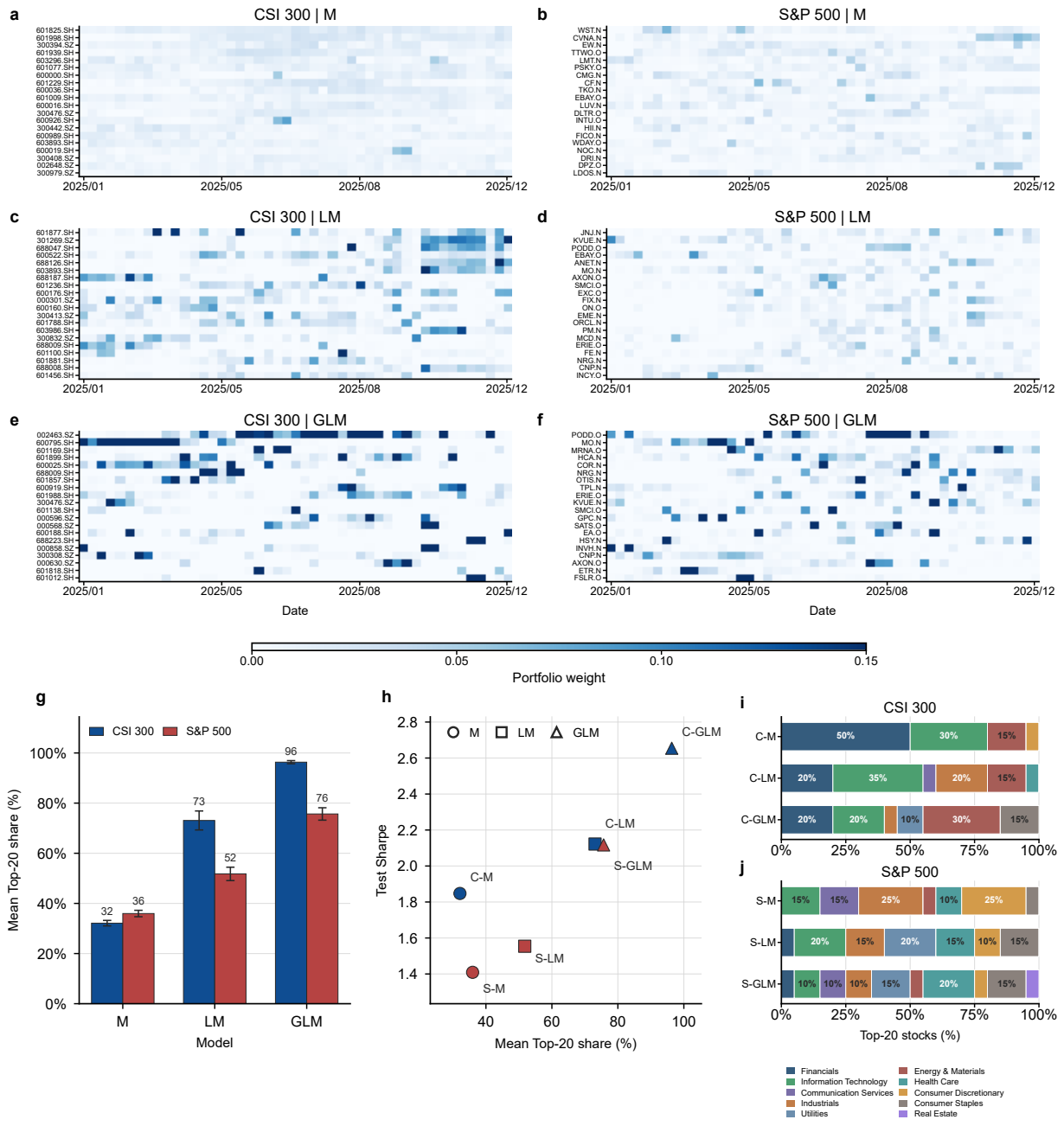


Fig. 3. Portfolio allocation patterns across markets and model architectures. **a**–**f**, Temporal allocation heatmaps. **a,c,e**, CSI 300 portfolios generated by M, LM, and GLM, respectively. **b,d,f**, Corresponding S&P 500 portfolios. Rows show the 20 stocks with the largest average test-period weights for each model. Columns represent rebalancing dates. Color intensity indicates portfolio weight assigned to each stock. **g**, Mean weight share allocated to the top 20 stocks across models and markets. **h**, Relationship between Sharpe ratio and mean top-20 weight share. C-M, C-LM, and C-GLM denote CSI 300 models; S-M, S-LM, and S-GLM denote S&P 500 models. **i,j**, Sector composition of top-20 stocks for CSI 300 and S&P 500, respectively. Each stacked bar shows the proportion of the 20 highest-weight stocks belonging to each industry category.

Table 3. Portfolio concentration and allocation diagnostics. Effective N is the reciprocal of the Herfindahl–Hirschman index and measures the effective number of holdings. Turnover quantifies the fraction of portfolio value traded between consecutive rebalancing dates. Weighted KL measures the Kullback–Leibler divergence between portfolio-weight distribution and uniform distribution across subsequent-return quantiles, indicating selectivity. Values are averaged over 47 test-period rebalancing dates. Standard deviations are shown in parentheses.

Dataset	Strategy	Effective N	Turnover	Weighted KL
CSI 300	M	108 (21)	0.29 (0.06)	0.04 (0.02)
	LM	26 (12)	0.59 (0.17)	0.25 (0.15)
	GLM	5 (3)	0.64 (0.22)	0.90 (0.38)
S&P 500	M	95 (19)	0.45 (0.08)	0.05 (0.02)
	LM	54 (17)	0.69 (0.13)	0.14 (0.10)
	GLM	18 (9)	0.71 (0.13)	0.31 (0.16)

The relationship between concentration and performance is shown in Fig. 3g,h. In the Chinese market, the mean top-20 weight share increases from 32% for M to 73% for LM and 96% for GLM. In the American market, the corresponding values increase from 36% to 52% and 76%. Quantitative diagnostics are provided in Table 3. In the Chinese market, the effective number of holdings (reciprocal of the Herfindahl–Hirschman index; see Methods) decreases from 108 for M to 26 for LM and 5 for GLM. In the American market, the effective number decreases from 95 to 54 and 18. Increasing portfolio concentration is accompanied by higher Sharpe ratios in both markets (Fig. 3h). Models incorporating broader network information allocate capital to fewer stocks while achieving superior risk-adjusted performance.

The sector composition of the top-20 stocks is shown in Fig. 3i,j. Despite substantial increases in concentration from M to LM and GLM, selected stocks remain distributed across multiple sectors in both markets. In the Chinese market, the M model concentrates 50% in Financials, with Information Technology (30%) and Energy & Materials (15%) accounting for the remainder. As network information is progressively incorporated, both LM and GLM reduce the Financials share from 50% to 20%. LM redistributes toward Industrials (20%) while adding Communication Services and Health Care, whereas GLM shifts weight to Energy & Materials (30%) and Consumer Staples (15%) and introduces Utilities (10%). A similar pattern occurs in the American market. The M model allocates substantially to Industrials and Consumer Discretionary (25% each), whereas GLM distributes more evenly across ten sectors. Network aggregation therefore does not concentrate capital at the sector level but broadens sector allocation. The observed performance improvements arise primarily from enhanced stock selection rather than sector exposure.

To quantify how portfolio weights are distributed across stocks with different subsequent returns, we compute the weighted Kullback–Leibler (KL) divergence (see Methods). Mean weighted KL divergence increases monotonically from 0.04 to 0.25 and 0.90 in the Chinese market, and from 0.05 to 0.14 and 0.31 in the American market. Higher weighted KL divergence indicates stronger differentiation of portfolio weights across subsequent-return groups, showing that broader network aggregation enables more selective capital allocation. Greater selectivity is accompanied by higher portfolio turnover. Average turnover increases from 0.29 to 0.59 and 0.64 across M, LM, and GLM in the Chinese market, and from 0.45 to 0.69 and 0.71 in the American market. The increase in turnover from LM to GLM is modest relative to the corresponding improvement in portfolio performance.

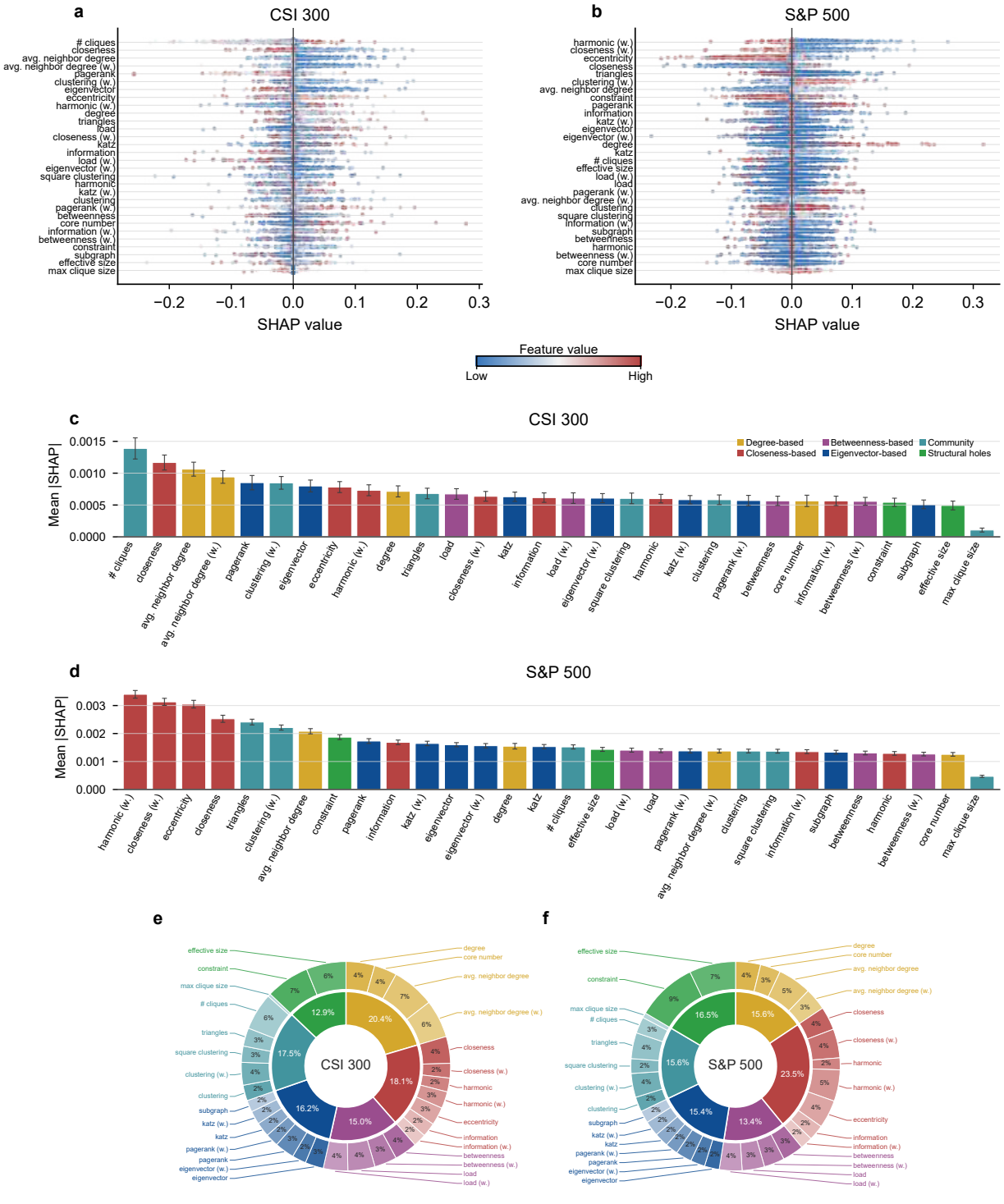


Fig. 4. Feature attribution analysis via SHAP for the GLM model. a,b, SHAP beeswarm plots for CSI 300 and S&P 500, respectively. Each point represents a single stock on a single rebalancing date. Color indicates the normalized feature value (red = high, blue = low). Features are ranked by mean absolute SHAP value. `avg.` denotes average; `(w.)` denotes weighted; `# cliques` denotes number of cliques. **c,d**, Mean absolute SHAP values with 95% confidence intervals for each feature. Colors denote topological feature classes (degree-based, closeness-based, and so on). **e,f**, Hierarchical sunburst visualization of feature importance. Inner ring shows six topological feature classes; outer ring shows 30 individual features. Percentages represent the normalized contribution to total mean absolute SHAP value. Labels identify class and feature names.

Feature importance

We apply SHAP analysis to identify which topological features are associated with portfolio weight variation in the GLM model (Fig. 4)[33]. The analysis covers all test-period observations: 14,029 in the Chinese market and 23,565 in the American market. For each stock, SHAP values quantify the contribution of individual topological features to the final portfolio weight while preserving the GNN message-passing and Transformer self-attention mechanisms.

SHAP beeswarm plots are shown in Fig. 4a,b, with features ranked from top to bottom by overall importance. In both markets, top-ranked features display a wide spread of SHAP values with substantial color mixing, indicating heterogeneous contributions across stocks and days rather than uniform directional effects. This pattern contrasts with bottom-ranked features (for example, *max clique size*), whose points cluster tightly near zero, reflecting minimal contributions.

Feature importance differs between markets, as shown in Fig. 4c,d. In the Chinese market, *number of cliques* is the most influential feature, followed by *closeness*, *average neighbor degree*, and *weighted average neighbor degree*. In the American market, distance-related features dominate. *Weighted harmonic centrality* ranks first, followed by *weighted closeness*, *eccentricity*, and *closeness*. Portfolio allocation in the Chinese market relies more heavily on local structure, whereas allocation in the American market depends more strongly on global reachability.

We aggregate the 30 topological features into six feature classes defined in Table 4. Class-level SHAP distributions (Fig. 4e,f) reinforce the distinction observed at the individual-feature level. In the Chinese market, degree-based (20.4%), closeness-based (18.1%), and community-related features (17.5%) account for the largest shares of total SHAP importance, highlighting the role of local structure. In the American market, closeness-based features contribute the largest share (23.5%), indicating stronger dependence on global reachability, although community and neighborhood features remain important.

Some topological features consistently receive little attribution in both markets. *Max clique size* ranks among the least important features in both markets (Fig. 4e,f). The empirical distribution shows that this feature is highly concentrated around a single value (Fig. 7 and Fig. 8). Most stocks have a max clique size of four, corresponding to the maximum clique size allowed by PMFG topology. Similarly, *core number* receives relatively low attribution in both markets and exhibits a highly skewed distribution. Topological features with limited structural variability contribute less to portfolio allocation.

Feature variability and its importance

Feature variability, quantified by the Shannon entropy of each feature's distribution across stocks and time, is compared with SHAP importance in Fig. 5. Feature entropy shows a positive linear relationship with SHAP importance in both markets, with a fitted slope of 0.019 for CSI 300 and 0.024 for S&P 500. In the Chinese market, *number of cliques* exhibits both high entropy and the highest attribution. In the American market, *weighted harmonic centrality* and *weighted closeness centrality* show large attribution shares alongside relatively high entropy. Conversely, *max clique size* and *core number* exhibit both low entropy and low attribution in both markets. These results indicate that features with greater distributional variability receive greater attribution in portfolio allocation decisions.

Feature attribution flow

We compare how SHAP attributions change across the M, LM, and GLM models to determine whether local or global aggregation merely amplifies existing signals or reorganizes underlying feature

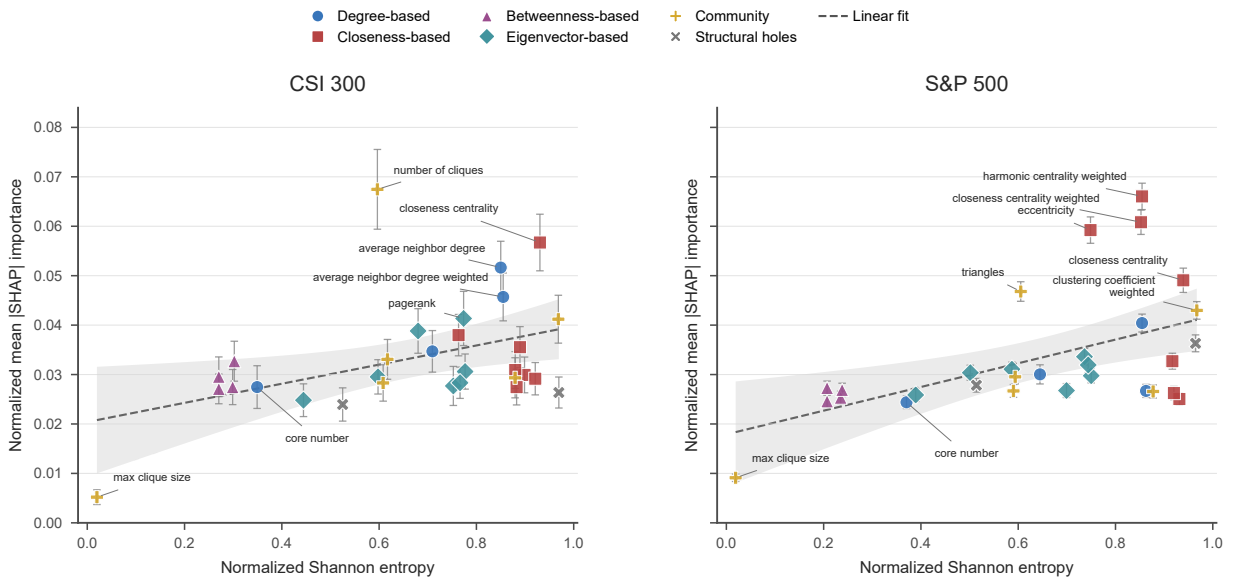


Fig. 5. Feature variability predicts attribution importance in the GLM model. Each point represents one of the 30 topological features. Color (or marker shape) indicates feature class. Fitted lines are from linear regression; shaded regions show 95% confidence intervals. Both Shannon entropy of the feature distribution and mean absolute SHAP importance are normalized within each market for comparability. Labels identify the five highest-importance features plus the features with extreme entropy values.

preferences. This attribution flow, shown in Fig. 6, describes the redistribution of feature importance as network information is progressively incorporated, quantified by changes in feature-importance ranking and attribution distribution across models. Rankings shift substantially from M to LM and GLM, indicating reorganization rather than simple reinforcement of the same features.

To assess how broadly this importance is distributed, we compute the Shannon entropy of the 30-feature SHAP-share distribution for each model. In both markets, entropy is highest for GLM, reaching 3.35 in CSI 300 and 3.34 in S&P 500, indicating that global self-attention redistributes attribution across a broader range of topological signals. This broader feature utilization coincides with more concentrated portfolio weights, suggesting that the model draws on a richer set of network information to make more selective allocation decisions. Local and global aggregation therefore do not simply amplify existing signals, they instead reorganize how different dimensions of market topology contribute to portfolio construction.

Discussion and conclusion

Portfolio optimization requires capturing dynamic cross correlations among assets, yet traditional mean-variance frameworks treat dependencies as static covariances. We develop a graph machine-learning framework that integrates dynamic network topology with portfolio construction, combining local neighborhood aggregation and global market dependencies through a hybrid GNN-Transformer architecture. Across Chinese and American stock markets, progressively richer network information enhances risk-adjusted returns while concentrating allocations. The GLM model, which leverages node-level features, local neighborhoods, and global dependencies jointly, achieves Sharpe ratios of 2.65 (CSI 300) and 2.12 (S&P 500), substantially outperforming both simpler architectures and market benchmarks.

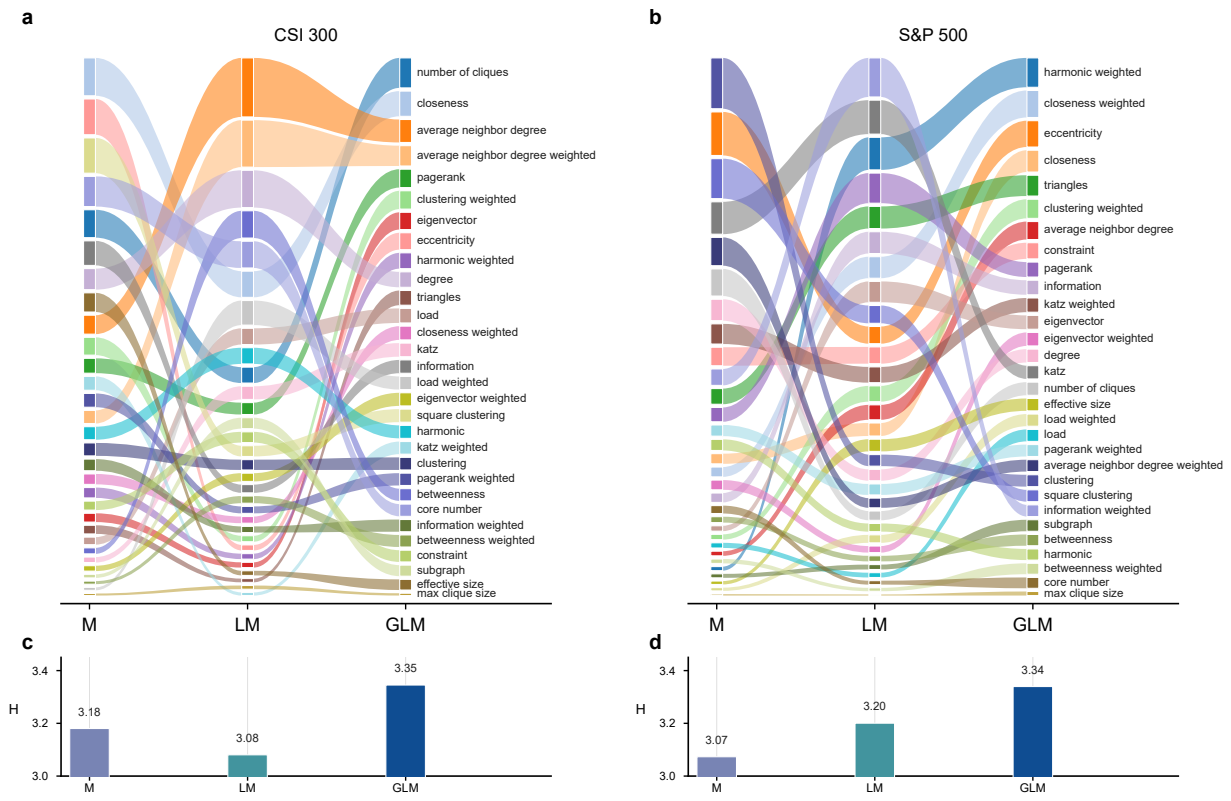


Fig. 6. Redistribution of feature importance across model architectures. Sankey diagrams show attribution flow from M to LM and GLM for **a**, CSI 300 and **b**, S&P 500. Node height and ribbon width represent the mean absolute SHAP importance share of each feature within the corresponding model. Features are ordered by their model-specific importance rank. **c,d**, Shannon entropy of the SHAP importance distribution across the 30 features. Higher entropy indicates more even distribution of attribution across features.

Portfolio allocation analysis reveals that richer network representations influence both concentration and feature utilization. As network information expands from node-level to local and global scales, portfolio weights become increasingly concentrated, whereas SHAP attribution distributes across a broader set of topological features. This dual pattern indicates that superior performance does not arise from a small number of dominant features but from integrating diverse structural information to form selective allocations. Despite increasing concentration, selected stocks remain distributed across industry sectors in both markets, indicating that performance gains stem from stock selection rather than sector tilts. The two markets exhibit distinct topological preferences: the Chinese market emphasizes local structural features, whereas the American market relies more on global reachability measures.

Feature variability proves predictive of attribution importance. Features with higher distributional entropy contribute more strongly to portfolio allocation, as quantified by SHAP values. This relationship holds across both markets, suggesting that network features carrying richer structural variation provide more informative signals for allocation decisions. Features constrained by topology (for example, max clique size limited by PMFG planarity) exhibit low variability and correspondingly low attribution.

The attribution flow analysis further shows that network aggregation reorganizes rather than merely amplifies feature importance. From M to LM and GLM, feature-importance rankings change

substantially. Shannon entropy of the SHAP-share distribution increases in GLM, indicating that global self-attention redistributes attribution across more features rather than concentrating it. This broader feature engagement occurs simultaneously with more selective portfolio weights, revealing that the model leverages a richer set of network signals to make more discriminating allocation decisions.

Despite these findings, several limitations remain. Financial markets generate information from multiple sources, such as price movements, macroeconomic factors and news. This study focuses exclusively on return-based correlation networks to isolate how much allocation-relevant information can be extracted from network topology alone. Integrating alternative data sources within the graph learning framework represents a natural extension. Moreover, although the Transformer module captures global dependencies among stocks, it does not explicitly model temporal dependencies within individual return series. Combining temporal graph learning with global aggregation could further enhance predictive performance. The PMFG filtering method retains a fixed fraction of edges, alternative filtering approaches may preserve different aspects of correlation structure and warrant investigation. Finally, transaction costs are not explicitly incorporated in the objective function, though turnover increases modestly from LM to GLM relative to performance gains.

Our results demonstrate that financial network topology provides interpretable signals for portfolio optimization and that graph machine learning can effectively integrate multiscale structural information. The consistent performance hierarchy across two markets with distinct characteristics highlights the generality of the approach. By translating network structure into constrained allocations through end-to-end learning, the framework moves beyond manually selected topological indicators toward a data-driven integration of node-level, local, and global information. This integration reveals that evolving correlation networks encode actionable dependencies that traditional methods overlook, opening pathways for network-informed asset allocation in diverse financial contexts.

Methods

Data

We collect daily closing prices for constituent stocks of the CSI 300 index (Chinese market) and the S&P 500 index (American market). The dataset spans 1 January 2022 to 31 December 2025 for both indices and comprises 300 stocks (CSI 300) and 503 stocks (S&P 500).

Network construction

Let $P_i(t)$ denote the closing price of stock i on trading day t. The logarithmic return is $R_i(t) = \ln P_i(t) - \ln P_i(t-1)$. For each day t, we compute the normalized return over a moving time window as

$$r_i(t) = \big(R_i(t) - \langle R_i \rangle_l\big)/\sigma_i(t), \tag{1}$$

where $\langle R_i \rangle_l = l^{-1} \sum_{\tau=0}^{l-1} R_i(t-\tau)$ is the mean return and $\sigma_i(t)$ is the standard deviation over the same window. We set $l = 125$ days throughout this study.

The cross-correlation matrix is defined by the equal-time correlation

$$C_{ij}(t) = \langle r_i(t)\, r_j(t) \rangle_l. \tag{2}$$

For each day t, we construct a network from the correlation matrix $C(t)$ using the planar maximally filtered graph (PMFG) method[19]. PMFG yields a sparse network by retaining the most informative links from the dense correlation network. For each stock in each daily network, we

compute 30 topological features spanning six categories (Table 4). The network structure and node features serve as inputs to the graph machine-learning model for portfolio allocation.

Table 4. Definitions and classification of the 30 node topological features used in this study. The 30 features are grouped into six categories[34,35]: degree-based centrality, closeness-based centrality, betweenness-based centrality, eigenvector-based centrality, community structure, and structural holes. Feature names ending in `_weighted` denote edge-weight-aware implementations. For weighted features, edge weights are transformed as $w_{ij}^{+} = \mathrm{clip}((w_{ij}+1)/2, 10^{-12}, 1)$ and $\delta_{ij} = 1 - w_{ij}^{+}$. Distance-based weighted features (closeness, betweenness, load, and harmonic centrality) use δ_{ij} as edge distances, whereas weighted PageRank uses w_{ij}^{+}. Weighted eigenvector centrality, Katz centrality, information centrality, clustering coefficients, and weighted average-neighbor degree use the original edge weights w_{ij}. All raw features are standardized within each daily graph before being used as model inputs.

Encoding Type	Feature Name	Definition
Degree-based centrality	`degree`	$D(i) = \frac{k_i}{n-1}$, where $k_i = \lvert N(i)\rvert$ is the degree of node i, $N(i)$ is its neighbor set, and $n = \lvert V\rvert$ is the number of nodes.
	`core_number`[36]	$\kappa(i) = \max\{r : i \in K_r(G)\}$, where $\kappa(i)$ is the core number, $K_r(G)$ is the r-core subgraph, and every node in $K_r(G)$ has degree at least r within that subgraph.
	`average_neighbor_degree`	$K_{nn}(i) = \frac{1}{k_i}\sum_{j\in N(i)} k_j$, where k_j is the degree of neighbor j.
	`average_neighbor_degree_weighted`	$K_{nn}^{w}(i) = \frac{1}{s_i}\sum_{j\in N(i)} w_{ij}k_j$, where $s_i = \sum_{j\in N(i)} w_{ij}$ is the weighted degree, w_{ij} is the edge weight.
Closeness-based centrality	`closeness`[37]	$C(i) = \frac{n-1}{\sum_{j\neq i} d(i,j)}$, where $d(i,j)$ is the unweighted shortest-path distance between nodes i and j, and n is the number of nodes.
	`closeness_weighted`[37]	$C^{\delta}(i) = \frac{n-1}{\sum_{j\neq i} d_{\delta}(i,j)}$, where $d_{\delta}(i,j)$ is the shortest-path distance formed from edge weights.
	`harmonic`[38]	$H(i) = \sum_{j\neq i} d(i,j)^{-1}$.
	`harmonic_weighted`[38]	$H^{\delta}(i) = \sum_{j\neq i} d_{\delta}(i,j)^{-1}$.
	`eccentricity`	$\mathrm{ecc}(i) = \max_{j\in V} d(i,j)$.
	`information`[39]	$\frac{\lvert V\rvert-1}{\sum_{u\neq v} r_{uv}}$, where r_{uv} is the resistance distance.
	`information_weighted`[39]	Weighted version of information centrality.
Betweenness-based centrality	`betweenness`[40]	$B(i) = \sum_{s\neq t\neq i} \frac{\sigma_{st}(i)}{\sigma_{st}}$, where σ_{st} is the number of shortest paths from s to t, $\sigma_{st}(i)$ is the number passing through i.
	`betweenness_weighted`[40]	$B^{\delta}(i) = \sum_{s\neq t\neq i} \frac{\sigma_{st}^{\delta}(i)}{\sigma_{st}^{\delta}}$, where σ_{st}^{δ} counts shortest paths under edge weights and $\sigma_{st}^{\delta}(i)$ counts those passing through i.
	`load`[41]	Total amount of packets passing through v when all node pairs exchange unit packets.
	`load_weighted`[41]	Weighted version of load centrality.
Eigenvector-based centrality	`eigenvector`[42]	$x_i = \lambda^{-1}\sum_{j\in N(i)} x_j$, λ is the leading eigenvalue of adjacency matrix A.
	`eigenvector_weighted`[42]	$x_i^{w} = \lambda^{-1}\sum_{j\in V} w_{ij}x_j^{w}$, w_{ij} is the edge weight.

Continued on the next page

Encoding Type	Feature Name	Mathematical Definition
	`pagerank`[43]	$PR_i = \frac{1-\alpha}{n} + \alpha \sum_{j \in N(i)} \frac{PR_j}{k_j}$, and the damping factor is $\alpha = 0.85$.
	`pagerank_weighted`[43]	$PR_i^w = \frac{1-\alpha}{n} + \alpha \sum_{j \in N(i)} \frac{w_{ji}^+}{\sum_{h \in N(j)} w_{jh}^+} PR_j^w$, w_{ij}^+ is the edge weight, and $\alpha = 0.85$.
	`katz`[44]	$\mathbf{x} = \alpha A\mathbf{x} + \beta\mathbf{1}$, $\mathbf{1}$ is the all-ones vector, and $\alpha = 0.1$, $\beta = 1$.
	`katz_weighted`[44]	$\mathbf{x}^w = \alpha W\mathbf{x}^w + \beta\mathbf{1}$, where $W = [w_{ij}]$ is the weighted adjacency matrix, and $\alpha = 0.1$, $\beta = 1$.
	`subgraph`[45]	$S(i) = \sum_{r=1}^{n} v_r(i)^2 e^{\lambda_r}$, $v_r(i)$ is the component of the r-th normalized eigenvector at node i, λ_r is its eigenvalue.
Community Structure	`clustering`[46]	$C_3(i) = \frac{2T(i)}{k_i(k_i-1)}$ for $k_i \geq 2$, and $C_3(i) = 0$ otherwise, where $T(i)$ is the triangle count.
	`clustering_weighted`[47]	$C_3^w(i) = \frac{1}{k_i(k_i-1)} \sum_{\substack{j,h \in N(i) \\ j \neq h}} (\widehat{w}_{ij}\widehat{w}_{ih}\widehat{w}_{jh})^{1/3}$, where $\widehat{w}_{ij} = w_{ij} / \max_{(u,v) \in E} w_{uv}$, w_{ij} is the edge weight.
	`square_clustering`[48]	$C_4(i) = \frac{\sum_{\{u,v\} \subseteq N(i)} q_i(u,v)}{\sum_{\{u,v\} \subseteq N(i)} [q_i(u,v) + a_i(u,v)]}$, where $q_i(u,v)$ counts common neighbors of u and v other than i, and $a_i(u,v)$ counts admissible two-hop alternatives in the denominator.
	`triangles`	$T(i) = \frac{1}{2}(A^3)_{ii}$, where $T(i)$ is the number of triangles containing node i.
	`number_of_cliques`	$Q(i) = \lvert\{C \in \mathcal{C} : i \in C\}\rvert$, where $Q(i)$ is the number of maximal cliques containing node i, and $\mathcal{C}$ is the set of maximal cliques.
	`max_clique_size`	$Q_{\max}(i) = \max_{C \in \mathcal{C}:\, i \in C} \lvert C\rvert$, where $Q_{\max}(i)$ is the size of the largest maximal clique containing node i.
Structural holes	`constraint`[49]	$\mathrm{Con}(i) = \sum_{j \in N(i)} \left(p_{ij} + \sum_{\ell \in N(i)} p_{i\ell} p_{\ell j}\right)^2$, where $p_{ij} = a_{ij} / \sum_{h \in N(i)} a_{ih}$ is the proportion of node i's interaction allocated to neighbor j, and $a_{ij} = 1$ for the unweighted graph.
	`effective_size`[49]	$E(i) = k_i - \frac{2t_i^{\mathrm{ego}}}{k_i}$, where $E(i)$ is effective size, and t_i^{ego} is the number of edges among the neighbors of i.

Graph machine-learning architecture

Inspired by hybrid graph machine-learning architectures[25–27], we design the GLM model for portfolio optimization to jointly capture local neighborhood topology and global market dependencies. At each parallel encoder layer, a GNN module and a Transformer self-attention module operate in parallel on the same node representation. The GNN branch captures local structural information through message passing, whereas the Transformer branch models global dependencies through self-attention. The local and global representations are fused through a two-layer multilayer perceptron.

For each trading day t, the input is a graph $G_t = (V_t, E_t)$ with a node-feature matrix $X_t \in \mathbb{R}^{N_t \times d}$, where V_t denotes the stock set, E_t denotes the edge set, $N_t = |V_t|$ is the number of nodes, and $d = 30$ is the node-feature dimension. For encoder layer ℓ, let $H_t^{(\ell)} \in \mathbb{R}^{N_t \times d}$ denote the node embeddings, with initial representation $H_t^{(0)} = X_t$. The layer-normalized representation is $\widehat{H}_t^{(\ell)} = \mathrm{LN}\left(H_t^{(\ell)}\right)$ [50].

Local and global representations are computed as

$$H_{\text{local},t}^{(\ell+1)} = \text{Dropout}\Big(\text{GNN}^{(\ell)}(\widehat{H}_t^{(\ell)}, E_t)\Big) + H_t^{(\ell)}, \tag{3}$$

$$H_{\text{global},t}^{(\ell+1)} = \text{Dropout}\Big(\text{MHA}^{(\ell)}(\widehat{H}_t^{(\ell)})\Big) + H_t^{(\ell)}. \tag{4}$$

Residual connections and dropout[51] are applied at every layer. In this work, we use three parallel encoder layers ($\ell = 0, 1, 2$).

The GNN branch updates each stock representation from its neighbors while retaining a separate self-feature pathway. The local aggregation operation is

$$\text{GNN}^{(\ell)}(H, E_t) = \sigma(\tilde{A}_t H W_{\text{nbr}}^{(\ell)} + H W_{\text{self}}^{(\ell)} + b^{(\ell)}), \tag{5}$$

where $\tilde{A}_t = D_t^{-1} A_t$, D_t is the degree matrix $(D_t)_{vv} = |\mathcal{N}_t(v)|$, $\mathcal{N}_t(v)$ denotes the neighbor set of node v in graph G_t, and A_t is the adjacency matrix. σ is the ReLU activation function[52], and $b^{(\ell)}$ is the bias term. The first term aggregates neighboring node features, and the second term preserves the current node representation.

The Transformer branch uses self-attention to capture long-range dependencies among stocks. The global aggregation operation is

$$\text{MHA}(\widehat{H}) = \text{softmax}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_k}}\right) V, \tag{6}$$

where $Q = \widehat{H} W_Q$, $K = \widehat{H} W_K$, and $V = \widehat{H} W_V$. The projection matrices W_Q, W_K, and W_V map the normalized representation to query, key, and value spaces, respectively. We use a single attention head per layer with $d_k = d_v = d$.

Local and global representations are fused by addition, $U_t^{(\ell+1)} = H_{\text{local},t}^{(\ell+1)} + H_{\text{global},t}^{(\ell+1)}$, and refined through a residual position-wise feed-forward block:

$$H_t^{(\ell+1)} = \text{Dropout}\left(\text{FFN}\left(\text{LN}(U_t^{(\ell+1)})\right)\right) + U_t^{(\ell+1)}. \tag{7}$$

The FFN uses ReLU activation, expands the hidden dimension to $2d$, and projects back to d.

The final node embedding $h_{i,t}$ of stock i on day t is passed to a multilayer perceptron (MLP):

$$z_{i,t}^{(0)} = h_{i,t}, \tag{8}$$

$$z_{i,t}^{(1)} = \text{Dropout}\Big(\sigma\left(W^{(1)} z_{i,t}^{(0)} + b^{(1)}\right)\Big), \tag{9}$$

$$s_{i,t} = W^{(2)} z_{i,t}^{(1)} + b^{(2)}, \tag{10}$$

where $s_{i,t}$ is the allocation score for stock i, and layer dimensions are $d \to d/2 \to 1$.

Allocation scores are transformed into long-only portfolio weights through temperature-scaled softmax:

$$w_{i,t} = \frac{\exp(s_{i,t}/T)}{\sum_{j \in V_t} \exp(s_{j,t}/T)}, \qquad w_{i,t} \geq 0, \qquad \sum_{i \in V_t} w_{i,t} = 1, \tag{11}$$

where T is the softmax temperature parameter. We set $T = 0.5$ for CSI 300 and $T = 0.3$ for S&P 500.

Baseline variants

We construct two baseline variants to isolate the contribution of different information scales. The node-feature MLP (M) computes each stock's allocation score directly from the node-feature vector X_t through the same MLP head, without local message passing or global self-attention. The local model (LM) retains the GNN encoder but removes the Transformer branch.

M, LM, and GLM use the same standardized node features, MLP head, long-only constraint, and market-specific temperature-scaled softmax. Their only architectural difference is the spatial range of information available to the encoder: node-level features for M, local messages for LM, and local plus global interactions for GLM.

Training objective and optimization

For each graph on date t, the model generates a portfolio allocation $\mathbf{w}(t) = (w_1(t), \ldots, w_{N_t}(t))$, where N_t is the number of stocks. The portfolio is then held for the subsequent m trading days.

The portfolio return on day t is

$$r_p(t) = \sum_{i=1}^{N_t} w_i(t) \left(\frac{P_i(t)}{P_i(t-1)} - 1 \right). \tag{12}$$

The cumulative holding-period return is

$$R_p(t+m) = \prod_{\tau=1}^{m} [1 + r_p(t+\tau)] - 1. \tag{13}$$

To account for risk, the volatility of the holding-period returns over the subsequent m trading days is defined as

$$\sigma_p(t+m) = \sqrt{\frac{1}{m-1} \sum_{\tau=1}^{m} \Big(r_p(t+\tau) - \bar{r}_p(t) \Big)^2}, \tag{14}$$

where $\bar{r}_p(t) = \frac{1}{m} \sum_{\tau=1}^{m} r_p(t+\tau)$ denotes the mean portfolio return over the holding period.

Finally, the loss function is defined as

$$\mathcal{L} = -\frac{1}{M} \sum_{t=1}^{M} \frac{R_p(t)}{\sigma_p(t)}, \tag{15}$$

where M is the number of training samples. In this study, the holding horizon is fixed at $m = 5$ trading days.

Training, validation, and test partitions are summarized in Table 1, while all hyperparameter settings are reported in Table 5. Validation is used for early stopping to mitigate overfitting. The same objective function is evaluated on the validation set. To ensure strict temporal separation, samples whose five-day holding periods crossed partition boundaries are excluded from the analysis.

Backtesting and performance metrics

The portfolio is rebalanced every five trading days. At each rebalance date t, portfolio weights are determined from the network constructed on that date and subsequently are held for the next five trading days. Over the test period, this procedure produces 47 portfolio rebalancing events.

On trading day t, the net asset value (NAV) is calculated as

$$\mathrm{NAV}(t) = \prod_{d=1}^{t} \left[1 + r_p(d)\right]. \tag{16}$$

The cumulative return is defined as

$$R_{\mathrm{cum}} = \mathrm{NAV}(t) - 1. \tag{17}$$

Assuming a zero risk-free rate, the Sharpe ratio is calculated using the arithmetic mean of daily returns:

$$\mathrm{Sharpe} = \sqrt{252}\,\frac{\langle r_p(t)\rangle}{\sigma}. \tag{18}$$

where $\langle ...\rangle$ means the average over time t, and σ denotes the standard deviation of daily returns. The Sortino ratio replaces the total volatility by the downside deviation computed from negative daily returns only.

Maximum drawdown (MDD) is defined as the largest peak-to-trough decline of the portfolio value during the backtest period,

$$\mathrm{MDD} = \min_{d} \left(\frac{\mathrm{NAV}(d)}{\max_{u \le d} \mathrm{NAV}(u)} - 1 \right). \tag{19}$$

Portfolio turnover between two consecutive rebalancing dates is measured as

$$\mathrm{Turnover}(t) = \frac{1}{2} \sum_{i \in V_t \cup V_{t-1}} \left| w_i(t) - w_i(t-1) \right|, \tag{20}$$

where absent stocks are assigned zero weight.

As a benchmark, we consider a market portfolio that assigns weight $1/N_t$ to every stock available on date t.

Portfolio concentration and allocation diagnostics

Portfolio concentration is evaluated at each trading day t using the Herfindahl–Hirschman index (HHI),

$$\mathrm{HHI}(t) = \sum_i w_i(t)^2, \tag{21}$$

where $w_i(t)$ denotes the portfolio weight assigned to stock i. The corresponding effective number of holdings is defined as

$$N_{\mathrm{eff}}(t) = \frac{1}{\mathrm{HHI}(t)}. \tag{22}$$

Larger values of N_{eff} indicate more diversified portfolios, whereas smaller values correspond to more concentrated allocations.

To quantify how strongly portfolio allocations deviate from a neutral allocation across realized return groups, we compute the Kullback–Leibler (KL) divergence. Stocks are first grouped into quantiles of equal size according to their subsequent k-day returns. Let $q_b(t)$ denote the fraction of stocks belonging to quantile b, and let $p_b(t) = \sum_{i \in S_b(t)} w_i(t)$ represent the portfolio weight allocated to that quantile. The KL divergence is then

$$D_{\mathrm{KL}}(t) = \sum_b p_b(t) \log \frac{p_b(t)}{q_b(t)}. \tag{23}$$

Higher KL values indicate that portfolio weights are more concentrated in specific return quantiles, whereas values close to zero indicate allocations similar to a market portfolio.

Table 5. Model training hyperparameters. All hyperparameter settings are shared across M, LM, and GLM within each market.

Hyperparameter	CSI 300	S&P 500
Architecture		
Input topology features	30	30
Parallel / MLP encoder layers	3	3
Parallel / MLP encoder channels	30	30
Self-attention heads	1	1
Activation	ReLU	ReLU
Optimization and regularization		
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Weight decay	10^{-5}	10^{-5}
Batch size	32	32
Dropout	0.05	0.10
Training schedule and allocation		
Training target horizon	5 trading days	5 trading days
Maximum epochs	120	150
Checkpoint eligibility (minimum epoch)	30	40
Early-stopping patience	40 epochs	50 epochs
Scheduler	ReduceLROnPlateau; factor 0.5; patience 5	ReduceLROnPlateau; factor 0.5; patience 5
Validation-loss smoothing window	5 epochs	5 epochs
Softmax temperature	0.5	0.3

SHAP analyses

To interpret the contribution of network topology to portfolio allocation, we apply SHAP to analyse the final portfolio-weight prediction of each stock[33]. For a given stock, the corresponding node-feature vector serves as the explanatory input, while the remaining graph structure and trained model parameters remain fixed. The complete model is re-evaluated for each perturbation, thereby preserving graph message passing, self-attention, and portfolio-weight normalization.

For a feature i, the SHAP value is defined as

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \Big[f(S \cup \{i\}) - f(S) \Big], \tag{24}$$

where F denotes the full feature set, S is a subset of features, and $f(\cdot)$ represents the model output. The SHAP value therefore measures the average marginal contribution of feature i across all possible feature subsets.

Feature importance is quantified by the mean absolute SHAP value,

$$I_i = \frac{1}{N} \sum_{n=1}^{N} \big| \phi_i^{(n)} \big|, \tag{25}$$

where N is the number of stock-day observations and $\phi_i^{(n)}$ is the SHAP value of feature i for observation n. Features are ranked by I_i.

To characterize how attribution is distributed across the $K = 30$ topological features, we normalize the importance values into a probability-like distribution, $q_i = \frac{I_i}{\sum_{k=1}^{K} I_k}$, and compute the Shannon

entropy of this distribution,

$$H = -\sum_{i=1}^{K} q_i \log q_i. \tag{26}$$

A higher H indicates that attribution is spread more evenly across features, whereas a lower H indicates that a small number of features dominate the total importance.

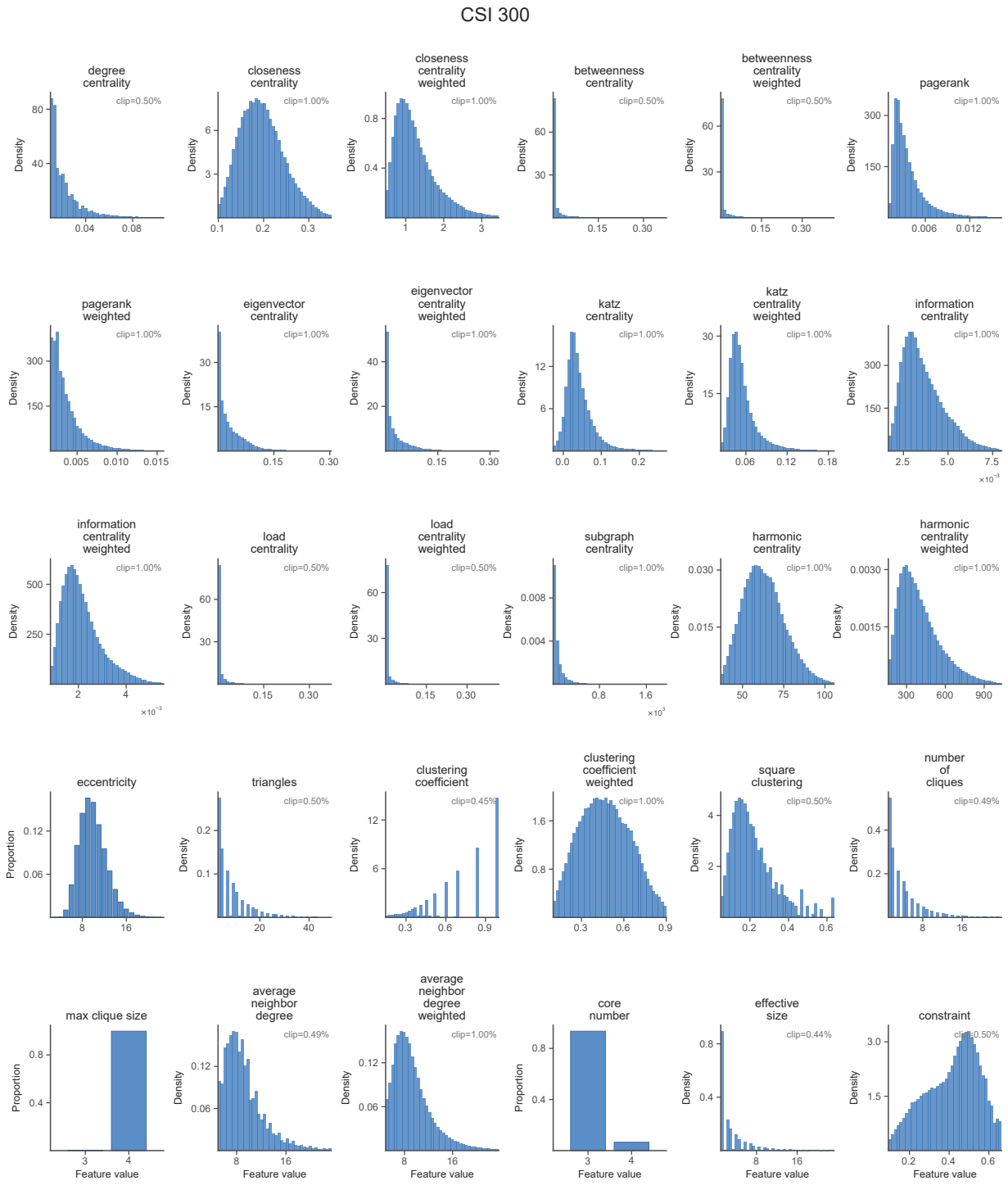


Fig. 7. Empirical distributions of topological features in the Chinese market. Each panel shows one of the 30 topology features across all available observations in CSI 300. Continuous features are plotted as densities after clipping values outside the 0.5th–99.5th percentile range for display.

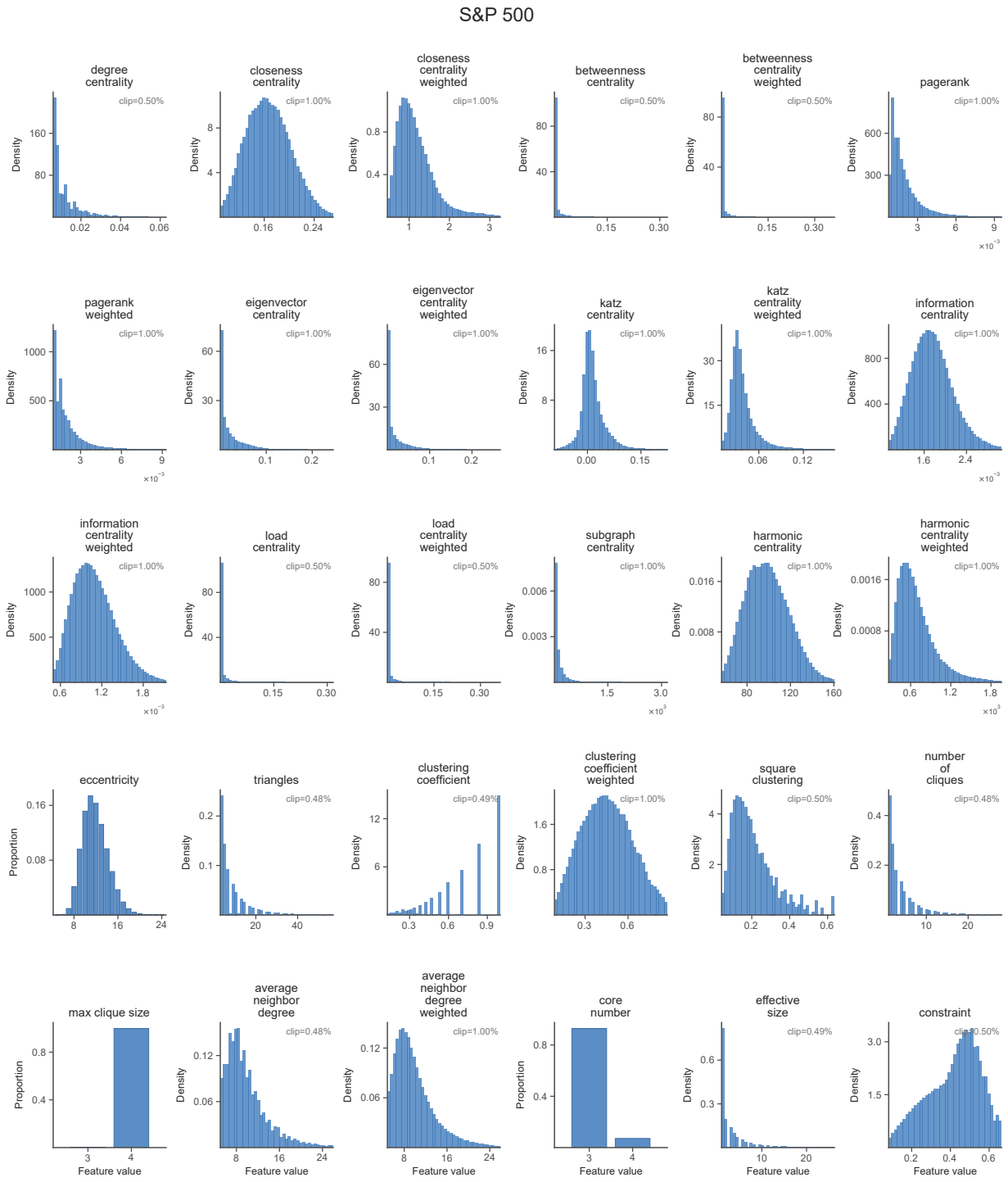


Fig. 8. Empirical distributions of topological features in the American market. Each panel shows one of the 30 topology features across all available observations in S&P 500. Continuous features are plotted as densities after clipping values outside the 0.5th–99.5th percentile range for display.

Data availability

The raw price data used in this study were obtained from Wind under a commercial licence and cannot be redistributed by the authors.

Acknowledgements

We thank Weihan Liu for helpful suggestions on this work. This work was supported by the Natural Science Foundation of Fujian Province (No. 2026J008332), the National Natural Science Foundation of China (No. 12305053), the Yunnan Fundamental Research Project (No. 202401CF070167), and the High-level Scientific Research Incubation Project of Ningbo University of Finance & Economics (No. 1320263705).

Author contributions

J.Z., L.X., and X.J. developed the concept, conceived and supervised the study. J.Z., L.X., X.J., and Y.L. designed and implemented the methodology. J.Z., L.X., X.J., and T.C. performed the validation and analysed the results. T.C. curated the data. All authors contributed to the writing and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

References

[1] Steven H Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.

[2] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47, 2002.

[3] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Phys. Rep.*, 424(4-5):175–308, 2006.

[4] Chaoming Song, Shlomo Havlin, and Hernan A Makse. Self-similarity of complex networks. *Nature*, 433(7024):392–395, 2005.

[5] Jessica Xin Hu, Cecilia Engel Thomas, and Søren Brunak. Network biology concepts in complex disease comorbidities. *Nat. Rev. Genet*, 17(10):615–629, 2016.

[6] Albert Lin, Runzhe Yang, Sven Dorkenwald, Arie Matsliah, Amy R Sterling, Philipp Schlegel, Szi-chieh Yu, Claire E McKellar, Marta Costa, Katharina Eichler, et al. Network statistics of the whole-brain connectome of drosophila. *Nature*, 634(8032):153–165, 2024.

[7] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, June 2005. ISSN 1476-4687.

[8] Dashun Wang, Chaoming Song, and Albert-László Barabási. Quantifying long-term scientific impact. *Science*, 342(6154):127–132, 2013.

[9] Santo Fortunato, Carl T. Bergstrom, Katy Börner, James A. Evans, Dirk Helbing, Staša Milojević, Alexander M. Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, Alessandro Vespignani, Ludo Waltman, Dashun Wang, and Albert-László Barabási. Science of science. *Science*, 359(6379):eaao0185, 2018.

[10] Romeil S. Sandhu, Tryphon T. Georgiou, and Allen R. Tannenbaum. Ricci curvature: An economic indicator for market fragility and systemic risk. *Science Advances*, 2(5):e1501495, 2016.

[11] Marco Bardoscia, Paolo Barucca, Stefano Battiston, Fabio Caccioli, Giulio Cimini, Diego Garlaschelli, Fabio Saracco, Tiziano Squartini, and Guido Caldarelli. The physics of financial networks. *Nature Reviews Physics*, 3:490–507, 2021.

[12] Matthew O. Jackson and Agathe Pernoud. Systemic risk in financial networks: A survey. *Annual Review of Economics*, 13:171–202, 2021.

[13] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[14] Hamidreza Esmalifalak and Amir Moradi-Motlagh. Correlation networks in economics and finance: A review of methodologies and bibliometric analysis. *Journal of Economic Surveys*, 39 (3):1252–1286, 2025.

[15] Lifu Jin, Bo Zheng, Xiongfei Jiang, Long Xiong, Jiu Zhang, and Jiahao Ma. Dynamic cross-correlation in emerging cryptocurrency market. *Physica A: Statistical Mechanics and its Applications*, 668:130568, 2025.

[16] Jukka-Pekka Onnela, Anirban Chakraborti, Kimmo Kaski, János Kertész, and Antti Kanto. Dynamics of market correlations: taxonomy and portfolio analysis. *Physical Review E*, 68(5): 056110, 2003.

[17] Tian Qiu, Bo Zheng, and Guang Chen. Financial networks with static and dynamic thresholds. *New Journal of Physics*, 12(4):043057, apr 2010.

[18] Rosario N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B*, 11(1):193–197, 1999.

[19] Michele Tumminello, Tomaso Aste, Tiziana Di Matteo, and Rosario N. Mantegna. A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences*, 102 (30):10421–10426, 2005.

[20] S. R. Nanda, Biswajit Mahanty, and Manoj Kumar Tiwari. Clustering indian stock market data for portfolio management. *Expert Systems with Applications*, 37(12):8793–8798, 2010.

[21] Fei Ren, Ya-Nan Lu, Sai-Ping Li, Xiong-Fei Jiang, Li-Xin Zhong, and Tian Qiu. Dynamic portfolio strategy using clustering approach. *PLoS ONE*, 12(1):e0169299, 2017.

[22] Francesco Pozzi, Tiziana Di Matteo, and Tomaso Aste. Spread of risk across financial markets: better to invest in the peripheries. *Scientific Reports*, 3:1665, 2013.

[23] Yan Li, Xiongfei Jiang, Yue Tian, Sai-Ping Li, and Bo Zheng. Portfolio optimization based on network topology. *Physica A: Statistical Mechanics and its Applications*, 515:671–681, 2019.

[24] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[25] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[27] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 14501–14515, 2022.

[28] Manali Patel, Krupa Jariwala, and Chiranjoy Chattopadhyay. A systematic review on graph neural network-based methods for stock market forecasting. *ACM Computing Surveys*, 57(2): 1–38, 2024.

[29] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems*, 37(2):1–30, 2019.

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

[31] Yi-Ling Hsu, Yu-Che Tsai, and Cheng-Te Li. FinGAT: Financial graph attention networks for recommending top-k profitable stocks. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):469–481, 2023.

[32] Chaoran Cui, Xiaojie Li, Chunyun Zhang, Weili Guan, and Meng Wang. Temporal-relational hypergraph tri-attention networks for stock trend prediction. *Pattern Recognition*, 143:109759, 2023.

[33] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[34] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[35] Stephen P. Borgatti and Martin G. Everett. A graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, 2006.

[36] Sergey N. Dorogovtsev, Alexander V. Goltsev, and José F. F. Mendes. K-core organization of complex networks. *Physical Review Letters*, 96(4):040601, 2006.

[37] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1 (3):215–239, 1978.

[38] Paolo Boldi and Sebastiano Vigna. Axioms for centrality. *Internet Mathematics*, 10(2–3): 222–244, 2014.

[39] Karen Stephenson and Marvin Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1–37, 1989.

[40] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1): 35–41, 1977.

[41] Kwang-Il Goh, Byungnam Kahng, and Doochul Kim. Universal behavior of load distribution in scale-free networks. *Physical Review Letters*, 87(27):278701, 2001.

[42] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[43] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[44] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[45] Ernesto Estrada and Juan A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Physical Review E*, 71(5):056103, 2005.

[46] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[47] Jari Saramaki, Mikko Kivela, Jukka-Pekka Onnela, Kimmo Kaski, and Janos Kertesz. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2): 027105, 2007.

[48] Pedro G. Lind, Marta C. González, and Hans J. Herrmann. Cycles and clustering in bipartite networks. *Physical Review E*, 72(5):056127, 2005.

[49] Ronald S. Burt. Structural holes and good ideas. *American Journal of Sociology*, 110(2):349–399, 2004.

[50] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[51] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[52] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.