# Graph Neural Networks for Multi-Modal Stock Trading with Reinforcement Learning Agent

Tianhui Huang, Haonan Zhu

October 20, 2025

## 1 Real-World Problem

Financial markets are highly interconnected. Traditional stock prediction models often treat each stock independently, ignoring inter-stock dependencies, sectoral relationships, supply chain links, and market sentiment. This leads to suboptimal forecasts and trading strategies. This project provides a graph-based, multi-modal approach to:

- Capture complex dependencies among stocks via heterogeneous edges (correlation, sector, fundamentals, supply chain, competitors, news sentiment).
- Generate interpretable embeddings identifying structural roles of stocks (hubs, bridges, role twins).
- Optimize sequential portfolio decisions by integrating learned embeddings into a reinforcement learning agent.

Impact: The framework improves prediction accuracy, financial interpretability, and portfolio performance, offering actionable insights for trading strategies in a structured, explainable manner.

## 2 Overview

We propose a **role-aware, multi-modal graph learning framework** for modeling interdependencies among stocks and optimizing trading strategies.

**Key highlights:**

- Nodes: stocks; edges: correlation, sector/industry, fundamental similarity, supply chain, competitors, news sentiment.
- Node features: technical indicators, fundamentals, sentiment, macro features.
- Graph Transformer + PEARL positional embeddings captures structural roles (hubs, bridges, role twins) for interpretable and robust embeddings.

- Reinforcement Learning (RL) agent uses embeddings to optimize sequential portfolio actions.
- Graph strategy: combined static + dynamic edges:
  - Static edges: sector, supply chain, competitor relationships (stable backbone)
  - Dynamic edges: rolling correlation, fundamental similarity, news sentiment (updated periodically)

This approach integrates predictive modeling, role-aware graph representation, and portfolio optimization, emphasizing interpretability and practical financial decision-making.

# 3 Input and Output

**Input:** Snapshot $G_t = (V, E_t, X_t)$ at each trading day $t$

- $V$ = stock nodes
- $E_t$ = edges capturing static and dynamic relationships
- $X_t$ = node features (technical, fundamental, sentiment, macro)

**Output:**

- Node-level prediction: 5-day-ahead stock return sign

$$y_{i,t+5} = \begin{cases} 1, & \text{if } \frac{p_{i,t+5} - p_{i,t}}{p_{i,t}} > 0 \\ 0, & \text{otherwise} \end{cases}, \quad \hat{y}_{i,t+5} = P(y_{i,t+5} = 1) \tag{1}$$

- RL agent action: Buy/Sell/Hold for each stock
- Evaluation metrics: accuracy/F1, IC, Precision@Top-K, backtested Sharpe ratio

# 4 Graph Construction

## 4.1 Static edges (stable backbone)

- Sector/Industry: binary or weighted
- Supply chain / competitor: binary

### 4.2 Dynamic edges (updated periodically)

**Rolling correlation:** Pearson correlation over 30-day window; connect two stocks if $|\rho_{ij}| > 0.6$

$$\rho_{ij,t} = \frac{\mathrm{cov}(r_i[t-30:t], r_j[t-30:t])}{\sigma_{r_i}\,\sigma_{r_j}}$$

- $r_i[t-30:t]$: log returns of stock $i$ over last 30 days
- $\sigma_{r_i}$: standard deviation of returns over window
- **Rationale:** 30-day window balances responsiveness and noise smoothing; threshold 0.6 captures meaningful co-movements while avoiding excessive sparsity

**Fundamental similarity:** cosine similarity of normalized metrics ([P/E, P/B, ROE, Debt, MarketCap]); connect if $> 0.8$

**Optional news sentiment edges:** weighted by sentiment score

**Combined Graph Strategy:**

- Static edges maintain stable clusters (sector, supply chain)
- Dynamic edges capture evolving market co-movements and cross-sector bridges
- **Rationale for Graph Transformer:** attention mechanism aggregates multi-edge, multi-relational information better than standard GCN/GAT

## 5 Node Features

- **Technical indicators:** returns (1-,5-,20-day), MA20/50, volatility, RSI, MACD, Bollinger Band width, ATR, OBV, trading volume, spikes
- **Fundamentals:** market cap, P/E, P/B, EV/EBITDA, ROE, EPS growth, debt/equity, current ratio, beta
- **Sentiment features:** news polarity, social media mentions
- **Macro/Supply chain:** sector index returns, interest rates, VIX, revenue exposure to major customers

## 6 Edge Types and Graphs

**Sector/Industry Description:** Stocks in the same sector or industry.
**Weight:** 1.0 (same industry), 0.5 (same sector).
**Rationale:** Ensures tight intra-industry clustering while maintaining broader sector-level relations.

**Rolling correlation Description:** Pearson correlation of returns over a 30-day window.
**Weight:** $\rho_{ij,t}$ if $|\rho_{ij,t}| > 0.6$.
**Rationale:** Captures meaningful co-movements while avoiding excessive noise; threshold chosen to control sparsity.

**Fundamental similarity Description:** Cosine similarity of normalized financial metrics (P/E, P/B, ROE, Debt, MarketCap).
**Weight:** $s_{ij}$ if $> 0.8$.
**Rationale:** Links stocks with highly similar financial profiles ("role twins") to capture comparable market behavior.

**Supply chain Description:** Customer → supplier relationships.
**Weight:** binary (1/0).
**Rationale:** Reflects revenue dependence; binary simplifies modeling of directional dependence.

**Competitor Description:** Competitor → competitor relationships.
**Weight:** binary (1/0).
**Rationale:** Ensures clear market adjacency and competitive links are modeled.

**News sentiment Description:** Stock → news relationships.
**Weight:** sentiment score.
**Rationale:** Weight proportional to sentiment intensity; optional. Adds dynamic market perception into the graph.

**Detailed explanation for rolling correlation:**

- Compute rolling 30-day log returns for each stock
- For every stock pair $(i, j)$, calculate Pearson correlation $\rho_{ij,t}$
- Only create an edge if $|\rho_{ij,t}| > 0.6$ to avoid noisy or insignificant connections
- The edge weight is set to $\rho_{ij,t}$, preserving positive/negative co-movement
- Updated periodically (e.g., every 20 trading days) to reflect evolving market dynamics

# 7 Model Architecture

- **Graph Transformer (Ying et al., 2021)**: Edge-aware global attention aggregates heterogeneous relationships; handles static + dynamic edges efficiently; 2 layers, 4 heads, hidden size 256
- **PEARL positional embeddings (You et al., 2023)**: Encodes structural roles (hubs, bridges, role twins); concatenated with node features
- **Rationale:** handles multi-relational, heterogeneous, combined static+dynamic edges; PEARL provides explainable structural roles; scales to larger subgraphs

## 8 RL Integration

- State: portfolio holdings + node embeddings + node features
- Action: Buy / Sell / Hold per stock
- Reward: portfolio return / risk-adjusted return
- Algorithm: Q-learning or policy gradient
- Optional: multi-agent RL managing subsets of portfolio

## 9 Pipeline

1. Data Collection & Feature Engineering: OHLCV, fundamentals, news sentiment; compute technical indicators, rolling correlations, fundamental similarity
2. Graph Construction: Build `HeteroData` graph with combined static + dynamic edges; assign edge weights; assign node features
3. Baseline GNN Training: `GCNConv`/`GATConv`/`SAGEConv` on static graph; demonstrates PyG layers, HeteroData handling
4. Graph Transformer + PEARL Training: edge-aware attention + role embeddings; node-level return prediction
5. Dynamic Graph Updates: update correlation/fundamental/news edges periodically; maintain static backbone
6. RL Integration & Portfolio Optimization: actions Buy/Sell/Hold; evaluate cumulative return, Sharpe ratio, drawdown
7. Evaluation & Visualization: quantitative (Accuracy, F1, IC, Precision@Top-K, Sharpe ratio); qualitative (t-SNE/UMAP on embeddings)

## 10 Evaluation

- Metrics: node-level (Accuracy, F1, IC, Precision@Top-K); portfolio-level (cumulative return, Sharpe ratio, max drawdown)
- Ablations: edge types (sector vs correlation vs fundamental); positional embeddings (fixed Laplacian vs PEARL); threshold sensitivity ($|\rho| \in [0.4, 0.8]$)
- Visualization: t-SNE/UMAP embeddings to interpret hubs, bridges, role twins

## 11 Milestones / MVP

| MVP | Description |
|---|---|
| 1 | Static stock-stock graph, technical features, GCN embedding, next-day return prediction |
| 2 | Multi-modal features + heterogeneous edges, Graph Transformer baseline |
| 3 | PEARL role-aware embeddings, end-to-end training |
| 4 | RL agent for sequential portfolio optimization, backtesting |
| 5 | *Optional:* Dynamic graph updates (combined static + dynamic) and multi-agent portfolio optimization |

## 12 Expected Contributions

1. Dataset: Dynamic multi-relational market graph (2015–2025) in PyG format

2. Model: Role-aware Graph Transformer + RL agent

3. Insights: Interpretable stock roles, cross-sector dependencies, evolving market structure

4. Performance: Improved predictive accuracy, financial interpretability, and trading results over traditional time-series or baseline GNNs

## 13 References


1. Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with GCNs. ICLR.

2. Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. NeurIPS.

3. Veličković, P., et al. (2018). Graph Attention Networks. ICLR.

4. Ying, R., et al. (2021). Do Transformers Really Perform Bad for Graph Representation? NeurIPS.

5. You, Y., et al. (2023). PEARL: Positional Embeddings for Role-Aware Graph Learning. NeurIPS.