Congressional Trading Prediction with Temporal Graph Networks

Project Chocolate: End-to-End Development of a TGN-Based Copy Trading System

FinTech Project Team

December 29, 2025

Agenda

Project Overview

Objective: Predict if copying a US Congressperson's stock trade will be significantly profitable (**1-month excess return vs SPY** $> 6\%$).

Why This Matters:

- Congress members have **potential informational advantages** (committee access, briefings).
- Their trades are **publicly disclosed** (STOCK Act, 2012).
- Prior research shows some congress members significantly outperform the market.

Core Challenge:

- Trades are **temporal** (patterns shift over time).
- Politicians have **interconnected behaviors** (trading the same stocks).
- Traditional tabular ML ignores these relational dynamics.

Why Temporal Graph Networks (TGN)?

Key Insight: Model the market as a **dynamic graph** where:

- **Nodes** = Politicians (∼500) and Stocks (∼2,000).
- **Edges** = Transactions (Buy/Sell events at specific timestamps).

Traditional ML

- Each trade is independent.
- No memory of past behavior.
- Static features only.

TGN Approach

- Trades are **connected**.
- **Memory** per node (track record).
- **Dynamic** features evolving over time.

Result: TGN can capture "streaks" and learn "who" trades "what" effectively.

Experiment Journey: From Baseline to Final Model

We conducted 13 systematic experiments to evolve the model:

1. **Exp 001-002**: Baseline TGN + Dynamic Label Masking
2. **Exp 003**: Two-Phase Training (major breakthrough)
3. **Exp 004-005**: Loss Function Experiments (Weighted BCE, Focal Loss)
4. **Exp 006**: Validation of Best Configuration
5. **Exp 007-010**: Market Context Features (OHLCV $\rightarrow$ Engineered)
6. **Exp 011**: Deep TGN (2-Layer GNN)
7. **Exp 012**: Deep Interaction Decoder
8. **Exp 013**: Scaling Experiments

Key Insight: Each component was tested independently to understand its contribution.

Exp 001-003: Establishing the Foundation

Exp 001: Baseline with Dynamic Labels

- Added `Masked_Label`: Historical trade outcome (Win/Lose) if resolved, else 0.5.
- Added `Age`: How old is each neighbor trade (log-normalized days).
- *Example:* 1 day old $\rightarrow \ln(1+1) \approx 0.69$, 10 days $\rightarrow \ln(1+10) \approx 2.40$.
- **Result**: High volatility (AUC: 0.42 – 0.81 across months).

Exp 002: Validation Split + Early Stopping

- 90/10 chronological split for validation.
- **Problem**: Holding out 10% of recent data hurt model quality.

Exp 003: Two-Phase Training $\Leftarrow$ Major Breakthrough

- Phase 1: Find optimal epoch count on 90% data.
- Phase 2: Retrain from scratch on 100% data for that many epochs.
- **Result**: AUC 0.548 $\rightarrow$ **0.601 (+9.7%)**

Exp 004-006: Loss Function Tuning

Exp 004: Weighted BCE + MeanAggregator

- Hypothesis: Class imbalance hurts performance.
- **Result**: Mixed – F1 improved (+0.029), but AUC dropped (-0.006).
- September collapsed (AUC 0.39).

Exp 005: Focal Loss ($\alpha = 0.25$, $\gamma = 2.0$)

- **Result**: Failed. All metrics degraded.
- Focal Loss too aggressive for this noisy dataset.

Exp 006: Validation Run

- Reverted to Unweighted BCE + MeanAggregator.
- **Confirmed**: Two-Phase Training is the key driver.
- AUC: 0.596, Macro-F1: 0.569. **Stable Baseline Established.**

Exp 007-010: Adding Market Context

Identified Gap: "The model has no access to market conditions."

Exp 007: Raw OHLCV Sequences (60-day)

- Added LSTM encoder for 60-day price history (Stock + SPY).
- **Result**: Performance dropped. LSTM introduced noise.

Exp 008: Filing Date Fix

- Switched date basis from Trade Date to Filing Date.
- **Result**: Stability improved (Macro-F1 +5%).

Exp 009-010: Engineered Features ⇐ Key Improvement

- Replaced LSTM with MLP on 14 engineered features:
- `Return_1d`, `Return_5d`, `Return_10d`, `Return_20d`, `Volatility_20d`, `RSI_14`, `Vol_Ratio` (x2 for Stock + SPY).
- Added `BatchNorm1d` + `Dropout(0.2)`.
- **Result**: Macro-F1 recovered to 0.560.

Exp 011-012: Deepening the Architecture

Exp 011: Deep TGN (2-Layer GNN)

- Single TransformerConv → 2-Layer Stack with ReLU + Dropout.
- **Result**: AUC 0.597 (New Best). More robust across months.

Exp 012: Deep Interaction Decoder ⇐ Final Architecture

- Old: `Sum(Src, Dst)` → `ReLU` → `Linear(1)`.
- New: `Concat(Src, Dst)` → `MLP(296→128→64→1)`.
- **Rationale**: Learn politician-stock *interactions* (e.g., "Pelosi + Tech").
- **Result**: **AUC 0.600** – Broke the 0.60 barrier!

Exp 013: Scaling Up (256 dims, 8 heads)

- **Result**: Overfitting. Performance regressed.
- **Conclusion**: Current capacity is sufficient.

Graph Construction

Data Representation: Continuous Temporal Bipartite Graph

Nodes:

- **Politicians** (Source): $\sim$500 unique.
- **Stocks** (Destination): $\sim$2,000 unique.

Edges (Events):

- Represent a single `BUY` or `SELL`.
- Timestamp: Filing Date.
- Directed: *Politician* $\rightarrow$ *Stock*.

Dataset:

- **Period:** 2012–2024 ($\sim$28,000 transactions).
- **Source:** Capitol Trades / House/Senate Stock Watcher.
- **Label:** Binary (1-Month **Post-Filing Excess Return** vs SPY $> 6\%$).

Feature Engineering (Complete)

1. Dynamic Edge Features (per Transaction):

Feature	Description
`Amount`	Log-normalized USD transaction size.
`Is_Buy`	Directional indicator (+1 Buy, -1 Sell).
`Filing_Gap`	Days between Trade and Disclosure.
`Masked_Label`	**Historical outcome** (Win/Lose) if resolved, else 0.5.
`Age`	Log-normalized days since neighbor trade.

2. Static Node Features: Party (8-dim), State (8-dim) embeddings.

3. Engineered Market Features (14 dims):

- `Return_1d, 5d, 10d, 20d`, `Volatility_20d`, `RSI_14`, `Vol_Ratio`.
- Computed for both target stock and SPY (market benchmark).

Key Anti-Leakage: `Masked_Label` only reveals resolved trade outcomes.

TGN Architecture (Final)

1. **Price Encoder (MLP):**
   - Input: 14 engineered features → Output: 32-dim embedding.
   - Includes `BatchNorm1d` + `Dropout(0.2)`.
2. **Memory Module (GRU):**
   - Each node maintains hidden state $h(t) = 100$-dim.
   - Message: `[Amount, Is_Buy, Gap, Price_Emb]` (35-dim).
   - Aggregator: `MeanAggregator` (handles concurrent trades).
3. **Graph Embedding (2-Layer TransformerConv):**
   - 4 attention heads per layer, `Dropout(0.1)`.
   - Edge features: `[Time_Enc, Msg, Masked_Label, Age]`.
   - **Definition:** `Time_Enc`: Relative time encoding $(t_{current} - t_{neighbor})$ to capture temporal distance.
4. **Deep Interaction Decoder (MLP):**
   - Input: `Concat(Z_src, Z_dst, S_src, S_dst, P_emb)` = 296-dim.
   - **Definitions:**
     - `Z`: Dynamic embeddings from graph (Politician, Stock).
     - `S`: Static embeddings (Politician: Party/State, Stock: Placeholder).
     - `P_emb`: Market Price Embedding (14 engineered features).
   - Architecture: 296 → 128 → 64 → 1.

Expanding Window Evaluation ("Level 2" Rigor)

Goal: Simulate real-world copy trading with no future information leakage.

For each Test Month:

1. **Train:** All data from 2012 to (Test Month - 1 month).
2. **Gap Phase:** Most recent month. Trades exist, but labels are **unknown**. Forward-only mode (update memory, no backprop).
3. **Test:** Predict trades in the target month.

Two-Phase Training (per month):

1. Phase 1: Train on 90% of data, find best epoch via early stopping (patience=5).
2. Phase 2: Retrain fresh model on 100% data for best_epoch epochs.

Metrics: ROC-AUC, PR-AUC, F1-Score, Macro-F1.

Ablation Study: Which Signal Matters?

Objective: Isolate the contribution of each feature group over 6 years (2019–2024).

Configurations Tested:

1. **Politician Signal Only (`pol_only`):**
   - Graph + Memory + Static Embeddings (Party, State).
   - **Zero out** all 14 market features.
2. **Market Signal Only (`mkt_only`):**
   - 14 Engineered Market Features.
   - **Zero out** Static Embeddings.
3. **Full Model (`full`):** All features active.

Scale: 72 months $\times$ 3 modes = 216 model training runs.

Ablation Results: Yearly F1 Score

Year	Pol Only	Mkt Only	Full
2019	0.290	0.286	0.210
2020	0.487	**0.538**	0.484
2021	0.419	**0.479**	0.451
2022	**0.523**	0.429	0.503
2023	**0.647**	0.618	0.600
2024	**0.673**	0.664	0.645
Mean	0.452	**0.471**	0.443

Observation: `mkt_only` achieves highest F1 (0.471) – market context helps **classification calibration**.

Ablation Results: Yearly ROC-AUC

Year	Pol Only	Mkt Only	Full
2019	**0.535**	0.501	0.523
2020	0.525	**0.530**	0.512
2021	**0.583**	0.570	0.564
2022	**0.565**	0.550	0.559
2023	0.599	**0.609**	0.593
2024	0.604	**0.608**	0.580
Mean	**0.560**	0.553	0.551

Observation: `pol_only` achieves highest AUC (0.560) – politician identity is a strong **ranking signal**.

Ablation Visualization (Trend)

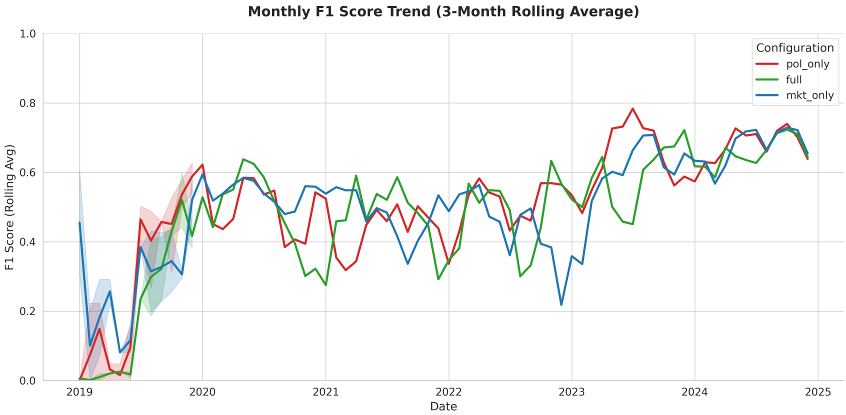

Figure: Monthly F1 Trend (3-Month Rolling Average). All models improve from 2019 to 2024.

Ablation Visualization

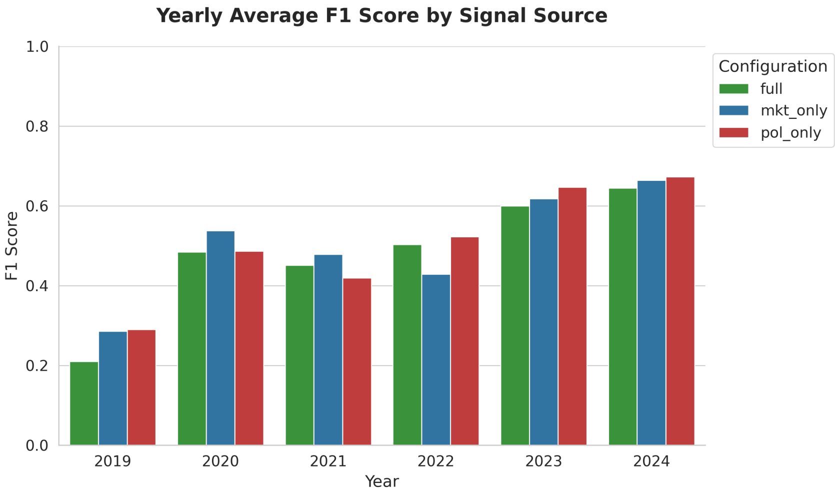

Figure: Yearly Average F1 Score by Signal Source (2019–2024).

Key Findings

1. **Two-Phase Training is Critical:**
   - Exp 003 showed +9.7% AUC improvement from this technique alone.
   - Using all recent data for backprop is essential.
2. **Engineered Features ¿ Raw Sequences:**
   - LSTM on raw OHLCV failed (Exp 007).
   - MLP on engineered features (RSI, Returns, Vol) succeeded (Exp 009-010).
3. **Politician Signal is Robust for Ranking:**
   - Ablation: Highest AUC (0.560) using only graph + static embeddings.
4. **Full Model Shows No Clear Synergy:**
   - Possible causes: Feature interference, overfitting, or suboptimal fusion.

Addressed Challenges & Iterations

We systematically addressed key modeling challenges:

Challenge	Status	Action
Memory underutilization	Fixed	Switched to MeanAggregator
Missing market context	Fixed	Added 14 engineered features
No validation set	Fixed	Two-phase training
Class imbalance	Partial	Tested Focal Loss (failed)
Extreme volatility	Partial	Improved but still present
Stock-stock relationships	Open	Future work

Limitations

- **Label Resolution Delay:** 1-month labels are noisy (short-term volatility).
- **Sample Size Variance:** Monthly counts range from 100 to 600. Small samples cause high AUC variance.
- **Feature Fusion:** Concatenation may not be optimal. Attention-based fusion unexplored.
- **Graph Structure:** No stock-to-stock relationships (sector correlations not captured).
- **Survivorship Bias:** Only trades from active members included.

Future Directions: Enhancing the Graph

1. **Stock-Stock Relationships:**
   - Add sector similarity edges.
   - Co-occurrence edges (stocks traded together).
2. **Politician Social Graph:**
   - Add **Politician-Politician edges** based on shared committee memberships.
   - Tests hypothesis: "Do committee peers trade similarly?"
3. **Dynamic Political Affiliation:**
   - Currently, Party is static (first observed).
   - Future: Attach Party as an **edge feature** to capture party switching or committee changes over time.

Future Directions: Model & Strategy

1. **Advanced Fusion Mechanisms:**
   - Cross-Attention between Static/Market branches.
   - Gated Fusion Networks.
2. **Politician Success Scores (Dynamic Feature):**
   - Historical win-rate per politician (updated monthly).
   - Leverage `BioGuideID` to track long-term performance.
3. **Copy-Trading Strategy Backtesting:**
   - Simulate portfolio returns based on model predictions.
   - Metrics: Sharpe Ratio, Max Drawdown, Calmar Ratio.
4. **Memory Decay Mechanism:**
   - Implement time-based decay for old memory states ($h(t)$).
   - Reduce influence of outdated trading patterns (e.g., ¿ 2 years).

Thank You!

Questions?

Project Directory: `/fintech/chocolate`