GNN-driven Macro Trading Strategy

Christine Wu
Carnegie Mellon University
88 Pine St, NY
cwu3@andrew.cmu.edu

Hansen Gan
Carnegie Mellon University
88 Pine St, NY
hgan@andrew.cmu.edu

Jacob Niyazov
Carnegie Mellon University
88 Pine St, NY
jniyazov@andrew.cmu.edu

Kevin Li
Carnegie Mellon University
88 Pine St, NY
kevinl4@andrew.cmu.edu

Tristan Roemer
Carnegie Mellon University
88 Pine St, NY
troemer@andrew.cmu.edu

Abstract

This project studies whether spatio-temporal graph neural networks (GNNs) can improve macro cross-asset trading by explicitly modeling dynamic, directional dependencies among key ETFs and derived ratio signals. Using a universe of 16 macro instruments spanning equities, rates/credit, commodities, FX, and volatility, we construct a time-varying asset graph and enrich node features with momentum, volatility, RSI, and discrete Lévy-area lead–lag statistics. We validate feature's predictive value through two transparent rule-based baselines: a long-only Lévy follower strategy versus an equal-weight market, and a targeted SPY/EFA overlay used for P&L attribution. The Lévy tests suggest economically meaningful follower outperformance under a positive leader shock, with the strongest results in a 50-day Lévy lookback and a 3-day holding horizon. Building on this, the GNN ensemble learns signed cross-asset edges and produces cross-sectional rankings that are mapped into a long-short portfolio with dynamic signal filtering. In preliminary backtests over 2010 to 2024, the GNN strategy delivers competitive risk-adjusted performance and materially reduced drawdowns relative to benchmarks. Overall, our results support a hybrid workflow in which Lévy-based lead-lag signals provide explainable directional priors and GNNs capture higher-order nonlinear propagation for regime-aware macro allocation.

1 Background and Introduction

Financial markets are constantly evolving, and the relationships between different asset classes rarely stay fixed for long. Equity indices, government bonds, credit spreads, commodities, currencies, and volatility all reflect different pieces of the macroeconomic environment. Under normal conditions, these instruments tend to move in recognizable patterns, i.e. bonds rally when equities sell off, commodities react to supply shocks, the dollar strengthens in risk-off environments, and so on. However, during periods of stress or after major policy surprises, these patterns can break down or even reverse. Having a model that can dynamically adapt to regime changes is critical for portfolio- and risk managers. A recent example is the Federal Reserve's rate cut, which had ripple effects across markets. Treasury yields fell immediately, equity indices rose, and volatility gauges like the VIX dropped as investors reassessed risk premia. At the same time, credit spreads and breakeven inflation adjusted in ways that reflected investors' forward-looking expectations. These cross-asset moves are not independent, they represent a network of interactions where some instruments consistently lead others. For instance, bond yields often react first to monetary policy, while equities and credit follow. Capturing these kinds of lead–lag effects is essential for building a reliable picture of how information diffuses through the system. Traditional time-series models, which typically analyze each instrument in isolation or assume static correlations, often miss the dynamic and directional nature of these relationships. They might capture average co-movement, but they are not designed to adapt when correlations spike or when the direction of influence flips. This creates blind spots for anomaly detection: an unusual move in one instrument may be ignored because the model doesn't "see" how it propagates to others. What is needed is a framework that can represent financial markets as a dynamic, interconnected system, where relationships evolve and shocks travel across assets in complex ways. Our project addresses this gap by exploring dynamic graph-based models. By representing each instrument as a node and the interdependencies as edges, we can model markets as a network that evolves over time. This allows us to detect not just isolated anomalies, but also systemic regime shifts where the network structure itself changes. For a firm, the ability to identify such changes are valuable in multiple contexts: (1) *Risk management:* spotting when normal hedging relationships (e.g., equities vs. Treasuries) break down; (2) *Trading:* detecting early when one market is leading another, creating opportunities or warning of contagion; and (3) *Policy analysis:* understanding how events like rate cuts or inflation surprises are transmitted across assets. Ultimately, the motivation is practical: firms need tools that go beyond static models to monitor and adapt to today's highly interconnected and non-stationary markets. By leveraging modern machine learning. Specifically, graph neural networks that can learn patterns in evolving graphs—we hope to develop a framework that captures the complexity of cross-asset dynamics and flags when something truly unusual is happening.

2 Motivation

Our work is motivated by a growing body of research arguing that cross-asset predictability is best understood through *network* structure rather than isolated time-series signals. In particular, *Network Momentum across Asset Classes* [1] demonstrates that momentum spillover can be captured using a learned, time-varying graph over a broad multi-asset universe. Their framework constructs network-weighted momentum representations and shows that these cross-asset features add predictive value beyond standard single-asset momentum indicators. The key takeaway for our setting is that a learned adjacency matrix can serve as an interpretable and compact summary of how momentum information diffuses across asset classes.

A complementary view is provided by *Follow the Leader: Enhancing Systematic Trend-Following Using Network Momentum* [2], which focuses on *directional* relationships. By detecting lead–lag structure and retaining only economically salient connections, the authors show that network-aware trend signals can improve traditional trend-following performance and downside characteristics. This emphasis on leadership and translation of lead–lag into actionable portfolio signals directly informs our use of Lévy-area features and our leader–follower baselines, which test whether directional cross-asset effects exist in our macro ETF universe before being embedded into the GNN.

Finally, the macro-finance motivation for time-varying network structure is reinforced by *Network connectedness and net spillover between financial and commodity markets* [3]. Using a rolling connectedness framework, this study shows that shock transmission across equities, bonds, currencies, and commodities is dynamic and economically interpretable, supporting the view that regime shifts often manifest as *changes in the network itself*. This provides an empirical rationale for moving beyond static correlations when designing signals intended for real-time portfolio decisions.

Together, these three strands motivate our core objective: to build a *regime-aware, context-dependent, and deployable* cross-asset signal engine for macro trading. We adopt the literature's insight that network structure can be learned from price-derived features, but extend it by using a spatio–temporal GNN to model nonlinear interactions and by explicitly incorporating Lévy-area lead–lag statistics as structured, directional information. Our goal is to deliver signals that are not only statistically credible but also operationally useful under realistic constraints on turnover, risk budgeting, and explainability.

3 Data Utilization

The dataset for this project is constructed from publicly available financial market instruments, sourced via the Yahoo Finance API (yfinance package in python). It spans multiple asset classes and provides a macro-level panel of features. The data covers the period from January 1st 2010, through to December 31st 2024.

3.1 Instruments and Indices

Our universe covers 16 key macro instruments and derived ratios, grouped into major asset classes. The dataset includes raw ETF price series and calculated spreads that capture specific risk premia (e.g., credit spreads, inflation expectations). All instruments are aligned on common trading dates, adjusted for dividends and splits, and cleaned for missing values using forward-fill for up to 5 business days.

Table 1 details the specific tickers and their economic roles within the model.

Table 1: Universe of Macro Instruments and Derived Ratios

Ticker	Instrument Name / Ratio	Economic Proxy
Equities		
SPY	SPDR S&P 500 ETF	U.S. Large-Cap
IWM	iShares Russell 2000 ETF	U.S. Small-Cap
EFA	iShares MSCI EAFE ETF	Developed Markets (ex-US)
EEM	iShares MSCI Emerging Markets	Emerging Markets
XLY/XLP	Discretionary / Staples Ratio	Cyclical vs. Defensive Sentiment
Rates & Credit		
SHY	iShares 1-3 Year Treasury Bond	Short-End Rates
IEF	iShares 7-10 Year Treasury Bond	Intermediate Rates (Duration)
LQD/IEF	Inv. Grade Corp / 7-10Y Treasury	Investment Grade Credit Spread
HYG/IEI	High Yield / 3-7Y Treasury	High Yield Credit Spread
TIP/IEF	TIPS / 7-10Y Treasury	Breakeven Inflation Expectations
Commodities		
USO	United States Oil Fund	Crude Oil Prices
GLD	SPDR Gold Shares	Gold Bullion
DBA	Invesco DB Agriculture Fund	Agricultural Commodities
Currencies & Volatility		
UUP	Invesco DB US Dollar Index	USD Strength
VIX	CBOE Volatility Index	Equity Implied Volatility
MOVE	Merrill Lynch Option Volatility Est.	Treasury Implied Volatility

4 Methods

We model multi-asset daily returns as a time-varying graph where each node represents an asset (ETF, index, or derived ratio series) and edges represent cross-asset dependencies learned by a Graph Attention Network. The system employs an ensemble of models with signed edge learning, differentiable rank correlation loss, and dynamic signal thresholding. For each trading day t, we construct a node-feature tensor from rolling windows of engineered indicators and Lévy-area lead-lag statistics. A temporal encoder compresses each asset's history into an embedding; signed-edge GATv2 layers perform cross-asset message passing; and a predictor head outputs cross-sectional scores for portfolio construction.

4.1 Feature Construction

For each asset i and day t, we compute the following features from daily closing prices:

- **Momentum Features:**

$$r_{1d}^{(i,t)} = \ln\left(\frac{P_t^{(i)}}{P_{t-1}^{(i)}}\right), \qquad r_{5d}^{(i,t)} = \ln\left(\frac{P_t^{(i)}}{P_{t-5}^{(i)}}\right), \qquad r_{20d}^{(i,t)} = \ln\left(\frac{P_t^{(i)}}{P_{t-20}^{(i)}}\right)$$

- 20-day rolling volatility, annualized by $\sqrt{252}$.
- **RSI (Relative Strength Index):** Computed with period 14, and rescaled to the symmetric range by

$$\frac{\text{RSI} - 50}{50}.$$

- **Lévy Area Lead–Lag Features:** For each ordered pair of distinct assets (i, j), we compute a rolling signed co-movement statistic (discrete Lévy area) summarizing directional influence. For a window of length W, given standardized price series x and y:

$$\mathcal{L}_{i,j}^{(t)} = \frac{1}{W-1} \sum_{k=1}^{W-1} \left(x_{k-1}\, \Delta y_k - y_{k-1}\, \Delta x_k\right).$$

  Positive values indicate asset i tends to *lead* asset j; negative values indicate the opposite.

- **Final Number of Features per Asset:** For N assets, each asset has $5 + (N-1)$ features, i.e., $N + 4 = 23$ when $N = 19$. All features are derived from *daily* close unless stated otherwise.

After construction, during training, all features are normalized with train-period z-scores.

This gives us a dataset that is a multi-asset, multi-feature daily time series panel that integrates equity, fixed income, credit, commodity, FX, and volatility dynamics. It consists of the daily historical data of 19 macro instruments, ratio series that capture cross-market spreads (such as equity cyclicals versus defensives or credit versus Treasuries), and volatility benchmarks. We engineer this historical data into 23 features for each asset, aiming to jointly provide a detailed description of both single-asset behavior and inter-asset dependencies. The data foundation ensures that future studies can replicate our results and obtain the same raw instruments directly from Yahoo Finance, as well as reconstruct the derived ratios and features with the pipeline described. To fully justify our inclusion of Levy Area as an informative feature in our GNN trading strategy, we have built a Long-Only Levy trading strategy with statistically significant signals that outperforms its benchmark, explained in the following section:

4.2 Lévy Trading Strategies

Before using the Lévy-area features inside the GNN, we first test whether they have standalone predictive content by building simple, fully rule-based strategies. Both strategies use the discrete Lévy area (following [4]) to identify *leaders* and *followers* in the cross-asset network.

Lévy matrix and scoring. Let $\mathcal{I}$ denote the set of N assets and let $r_t^{(i)}$ be the simple daily return of asset $i \in \mathcal{I}$ on day t. For a given lookback window W, we take the last W days of returns up to (but excluding) day t,

$$\mathcal{W}_t = \{r_\tau^{(i)} : \tau \in [t-W, t),\ i \in \mathcal{I}\},$$

standardize each asset within the window, and compute the discrete Lévy-area matrix $\mathbf{L}_t \in \mathbb{R}^{N \times N}$, whose (i, j) entry $\mathcal{L}_{i,j}^{(t)}$ is the Lévy area between assets i and j over $\mathcal{W}_t$. By construction, $\mathbf{L}_t$ is antisymmetric and we set the diagonal to zero.

To summarize the *outgoing* lead strength of each asset i, we form a non-negative score

$$s_i(t) = \sum_{j \in \mathcal{I},\, j \neq i} \max\big(\mathcal{L}_{i,j}^{(t)},\, 0\big), \qquad i \in \mathcal{I}. \tag{1}$$

Intuitively, $s_i(t)$ aggregates all positive Lévy mass flowing from i to other assets; assets with high scores are interpreted as *leaders*, while assets with low scores are *followers*.

Let $Q_t(\cdot)$ denote the empirical quantile of $\{s_i(t)\}_{i \in \mathcal{I}}$ and fix a percentile $p \in (0, 0.5)$ (e.g., $p = 20\%$). We define leader and follower sets on day t by

$$\mathcal{L}_t = \{i \in \mathcal{I} : s_i(t) \geq Q_t(1-p)\}, \tag{2}$$

$$\mathcal{F}_t = \{i \in \mathcal{I} : s_i(t) \leq Q_t(p)\}. \tag{3}$$

We then use the average same-day leader return,

$$\bar{r}_L(t) = \frac{1}{|\mathcal{L}_t|} \sum_{i \in \mathcal{L}_t} r_t^{(i)}, \tag{4}$$

as a scalar signal indicating whether the leading assets have just experienced a positive or negative shock. The graphs below illustrate the process of forming the pair-wise Levy's area matrix, keeping only the positive entries, and summing over the rows for a signal.

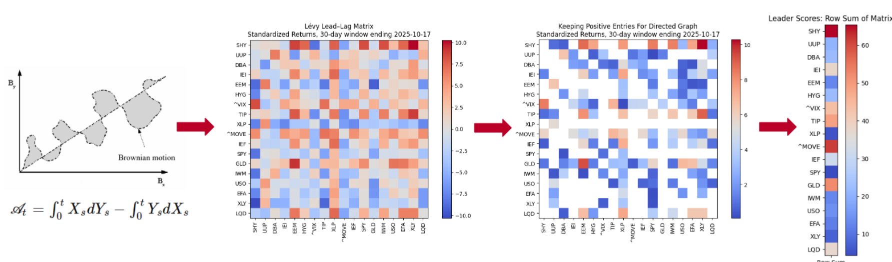

Figure 1: A graph example of constructing a leader score signal from Levy's area.

4.2.1 Strategy 1: Long-Only Lévy Followers vs. Market

Strategy 1 is a long-only, block-based strategy that switches between a portfolio of followers and the equal-weighted market based on the sign of $\bar{r}_L(t)$ in (4). We discretize time into non-overlapping blocks of length k trading days (the forecast horizon). A block starting at decision date t corresponds to the interval $(t, t+k]$.

For a block starting at t, we define:

- **Follower portfolio daily return** on day $\tau \in (t, t+k]$:
$$r_F(\tau; t) = \frac{1}{|\mathcal{F}_t|} \sum_{i \in \mathcal{F}_t} r_\tau^{(i)}.$$
- **Equal-weight market daily return** on day τ:
$$r_M(\tau) = \frac{1}{|\mathcal{I}_\tau|} \sum_{i \in \mathcal{I}_\tau} r_\tau^{(i)},$$
where $\mathcal{I}_\tau \subseteq \mathcal{I}$ is the set of assets with valid prices on day τ.

We then compound daily returns to obtain k-day block returns:

$$R_F^{(k)}(t) = \prod_{\tau=t+1}^{t+k} \big(1 + r_F(\tau; t)\big) - 1, \tag{5}$$

$$R_M^{(k)}(t) = \prod_{\tau=t+1}^{t+k} \big(1 + r_M(\tau)\big) - 1. \tag{6}$$

The Lévy follower strategy chooses between these two legs according to:

$$R_{\text{Lev},1}^{(k)}(t) = \begin{cases} R_F^{(k)}(t), & \text{if } \bar{r}_L(t) > 0, \\ R_M^{(k)}(t), & \text{otherwise.} \end{cases} \tag{7}$$

Blocks are non-overlapping: after using day t as an entry date, the next decision time is $t+k$. This ensures that each realized k-day return is based on information strictly prior to the block and helps simplify the statistical analysis of block-level performance.

We build on the open-source cryptocurrency lead–lag portfolio framework of Rahimi [5], which uses Lévy-area-based scores to identify leader and follower assets and to construct long-only and market-neutral portfolios. While their implementation focuses on cryptocurrencies, we adapt the same core ideas to a universe of macro ETFs and modify the portfolio construction rules to suit our setting. Figure 2 reproduces their schematic follower–leader trading logic.

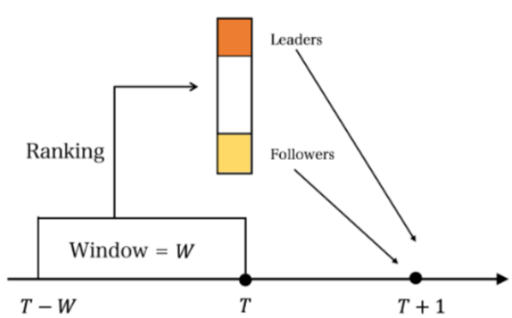

Figure 2: Demonstration of the follower–leader trading logic (adapted from Rahimi [5]).

Rationale. The Lévy matrix $\mathbf{L}_t$ encodes directional co-movement: $\mathcal{L}_{i,j}^{(t)} > 0$ suggests that shocks in asset i tend to precede shocks in asset j. Aggregating the positive outgoing Lévy mass via (1) identifies a small set of *leaders* and a complementary set of *followers*. When leaders experience a positive shock on day t (i.e., $\bar{r}_L(t) > 0$), we interpret this as new information entering the system through the leading nodes and expect it to diffuse towards followers. Strategy 1 therefore goes long the followers over the next k days only when the leader shock is positive, and otherwise defaults to the broad equal-weight market. This design keeps the strategy long-only, limits turnover, and isolates whether the Lévy-based ranking has predictive content relative to a simple market benchmark.

For significance, we focus on blocks where $\bar{r}_L(t) > 0$ and compare follower and market block returns via

$$\Delta^{(k)}(t) = R_F^{(k)}(t) - R_M^{(k)}(t),$$

computing sample means, t-statistics, bootstrap confidence intervals, and cross-sectional information coefficients between $-s_i(t)$ and k-day forward returns. The negative sign reflects that lower scores correspond to followers, which we expect to outperform conditional on a positive leader shock. These statistics are later shown in result section 5.1. We found that with a window of **50 days** to calculate Levy's area, and a holding periods of **3 days**, along with a top candidate percentage of **p=0.2, top 3 and bottom 3 assets**, gives the highest return and Sharpe. A very short window (e.g. 10–20 days) would make leaders highly sensitive to transient noise and reduce the effective sample size for estimating standardized co-movements across 19 instruments. Conversely, an excessively long window (e.g. > 100 days) would blur regime changes and mix together distinct volatility and correlation environments, leading to stale lead–lag structure. A 50-day window provides enough observations to estimate the standardized returns and Lévy areas with reasonable robustness. For $k = 3$, the mean excess follower block return versus the equal-weight market,is positive and economically meaningful (mean $\approx 0.21\%$ per 3-day block) with an approximate t-statistic of 2.19, indicating statistical significance at conventional levels. The hit rate (fraction of blocks where followers beat the market) is above 50%, and the average cross-sectional information coefficient between $-s_i(t)$ and forward 3-day returns is also positive (IC ≈ 0.033), suggesting that the Lévy-based ranking contains genuine predictive content. For other horizons, excess follower performance remains positive but exhibits weaker t-statistics or less stable ICs, often due to a smaller number of blocks (e.g. $k = 7, 10$) or higher noise ($k = 1$).

4.2.2 Strategy 2: Targeted SPY/EFA Levy Overlay

Strategy 2 is a more concentrated variant that uses the same Lévy infrastructure but only takes active risk in a pre-specified set of "positive" assets. Let $\mathcal{P} \subset \mathcal{I}$ denote this set; in our experiments, $\mathcal{P} = \{\text{SPY}, \text{EFA}\}$, corresponding to U.S. and developed ex-U.S. equities. At each decision date t we again compute $\mathbf{L}_t$, $s_i(t)$, $\mathcal{L}_t$, and $\mathcal{F}_t$ as above, and use the same leader shock $\bar{r}_L(t)$ from (4). We then define the subset of "positive" assets that currently sit in the follower tail as $\mathcal{P}_t = \mathcal{F}_t \cap \mathcal{P}$. For a fixed horizon k (e.g. $k = 5$ trading days), the block return of Strategy 2 starting at t is defined as

$$R^{(k)}_{\text{Lev},2}(t) = \begin{cases} R^{(k)}_{\mathcal{P}_t}(t), & \text{if } \bar{r}_L(t) > 0 \text{ and } |\mathcal{P}_t| > 0, \\ R^{(k)}_M(t), & \text{otherwise,} \end{cases} \tag{8}$$

where

$$R^{(k)}_{\mathcal{P}_t}(t) = \prod_{\tau=t+1}^{t+k} \left(1 + \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} r^{(i)}_\tau \right) - 1$$

is the k-day compounded return of an equal-weight portfolio of those positive assets that happen to be classified as followers at time t. If no positive asset lies in $\mathcal{P}_t$ or if the leader shock is non-positive, the strategy falls back to the equal-weight market block return $R^{(k)}_M(t)$. We primarily use Strategy 2 as a *P&L attribution tool* to identify which assets contribute most to the Lévy-based alpha. In our backtests, SPY and EFA stand out as followers with *statistically significant* excess returns when $\bar{r}_L(t) > 0$, which motivates their inclusion in $\mathcal{P}$. By contrast, most fixed-income ETFs exhibit overall positive performance and hit rates above 50% when acting as followers, but their excess follower–market returns are not statistically significant once we account for sampling variability. This suggests that the strongest and most robust lead–lag effects in our macro universe arise in equity indices rather than in duration or credit exposure. The logic is detailed in Algorithm 1.

Algorithm 1 Targeted Equity Lead–Lag Strategy

Input: Universe prices P, Lookback $W = 30$, Horizon $H = 7$, Targets $\mathcal{T} = \{\text{SPY, EFA}\}$
Initialize: $t \leftarrow W$
while $t < T - H$ **do**
  Step 1: Data Prep
  Extract window $P_{t-W:t}$
  Compute standardized returns Z
  Step 2: Lévy Matrix Construction
  Compute antisymmetric matrix L where $L_{ij} = \text{LevyArea}(Z_i, Z_j)$
  $S_i \leftarrow \sum_j \max(0, L_{ij})$ ▷ Calculate Leadership Scores
  Step 3: Regime Identification
  $\mathcal{L}_t \leftarrow$ Top 20% assets by S
  $\mathcal{F}_t \leftarrow$ Bottom 20% assets by S
  $\bar{r}_L \leftarrow$ Mean return of $\mathcal{L}_t$ on day t
  Step 4: Signal & Intersection Check
  $\mathcal{I}_t \leftarrow \mathcal{F}_t \cap \mathcal{T}$ ▷ Check if SPY/EFA are followers
  if $\bar{r}_L > 0$ **and** $\mathcal{I}_t \neq \emptyset$ **then**
    Position: Long Equal-Weight on $\mathcal{I}_t$ for $t + 1 \ldots t + H$
  else
    Position: Long Equal-Weight on Universe (Market) for $t + 1 \ldots t + H$
  end if
  $t \leftarrow t + H$ ▷ Move to next non-overlapping block
end while

Rationale. While Strategy 1 asks whether Lévy-based follower sets outperform the market on average, Strategy 2 targets a more realistic use case: timing exposure to a small set of macro-relevant indices. When SPY and/or EFA appear in the follower tail *and* the leaders have just rallied, the network suggests that equity risk is likely to benefit from the diffusion of that positive shock. In those cases, the strategy takes a concentrated long position in SPY/EFA; otherwise it simply holds the broad equal-weight market. This construction tests whether the Lévy signal provides value-add specifically for timing equity beta, while keeping the strategy long-only and capacity-friendly.

4.3 GNN Trading Strategy

We model our asset dynamics time-varying graph. Each node is an asset (ETF, index or ratio), and edges represent potential cross-asset dependencies learned by using a Convolutional Graph Attention Network (GAT). For each day t, we build a node-feature tensor from rolling windows of engineered indicators (returns, volatility, RSI) and Lévy-area lead-lag statistics that summarize directional influence between assets. A temporal encoder (bi-LSTM) compresses each asset's recent history into an embedding; a GAT layer stack performs cross-asset message passing; and a predictor head outputs cross-sectional scores used for both training (pairwise ranking loss) and portfolio formation (top/bottom selection). We evaluate the strategy in a walk-forward backtest with rolling H-day capital management (default H=5) and compare to an equal-weighted benchmark. We use a lookback window L=60 trading days for features. Chronological splits are approximately 70% train, 15% validation, 15% test (the final H days are excluded from modeling to form forward returns). Z-score normalization parameters (mean, standard deviation) are computed on the training slice (across time and assets) and applied unchanged to validation and test.

4.3.1 Model Architecture

The model architecture consists of four learnable components plus a prediction head:

1. **Temporal Encoder using LSTM:** Each asset's reweighted sequence (length 60) is passed through a bidirectional LSTM (e.g., hidden size 64–128, 2 layers, dropout). We concatenate the final forward and backward hidden states and project them to a fixed embedding size, yielding one temporal embedding per asset per day that summarizes recent dynamics.
2. **Signed Edge Network:** This network represents the edges of the graph and learns non-linear relationships between assets. The edges are initially inferred from the embeddings produced by the LSTM and later refined by the GNN. The architecture is:

$$\text{Linear}(2d \to d) \to \tanh \to \text{Linear}(d \to d/2) \to \tanh \to \text{Linear}(d/2 \to 1) \to \tanh$$

3. **Graphical Neural Network for encoding informative relationships:** We build a fully connected graph over assets (self-loops omitted) and apply two layers of GATv2Conv. The first GNN layer uses the edge weights learned by the Signed Edge Network to determine how much attention each asset should pay to incoming messages from other assets. It produces a rich embedding that encodes structural relationships across assets.

   This embedding is then used to update edge weights before being passed to the second GNN layer. While the first layer captures direct neighbor relationships, the second layer encodes neighbor-of-neighbor relationships. To preserve all information learned from both layers, we concatenate their embeddings (rather than using only the output of the second layer), which empirically improves performance.

4. **Prediction Head (MLP) for producing signal strengths:** The GNN output embedding for each asset is passed to a small Multi-Layer Perceptron (3 linear layers with ReLU and dropout). This head generates a scalar signal score for each asset at each time step corresponding to predicted relative over/underperformance.

4.3.2 GNN Training Objective and Procedure

Our training objective is to rank assets against each other by their 5-day forward returns rather than to predict exact returns. Our loss function is a differentiable rank correlation loss: for all asset pairs on a given day, we penalize cases where the predicted relative ordering disagrees with the realized ordering by more than a small margin.

Our ranking loss is defined as:

$$\mathcal{L}_{rank} = -\frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \tanh\left(\frac{\hat{y}_i - \hat{y}_j}{\tau}\right) \cdot \tanh\left(\frac{y_i - y_j}{\tau}\right)$$

To stabilize learning and discourage trivial constant outputs, we add a low-weight auxiliary MSE between cross-sectionally standardized predictions and standardized targets. Mini-batches are formed by sampling calendar dates from the train period; each batch assembles the asset × lookback × feature tensor and the aligned forward returns. We optimize with AdamW (typical learning rate 1e-3, weight decay 1e-4), apply a plateau-based learning-rate scheduler, and use early stopping on a validation ranking metric (Spearman rank/information coefficient). Random seeds are fixed for NumPy and PyTorch. Hyperparameters (lookback 60, horizon 5, Lévry window 20, LSTM hidden size, number of GATv2 layers and heads, dropout, learning rate, batch size, scheduler patience) are logged and kept constant across runs unless explicitly varied.

The figure below shows our model architecture:

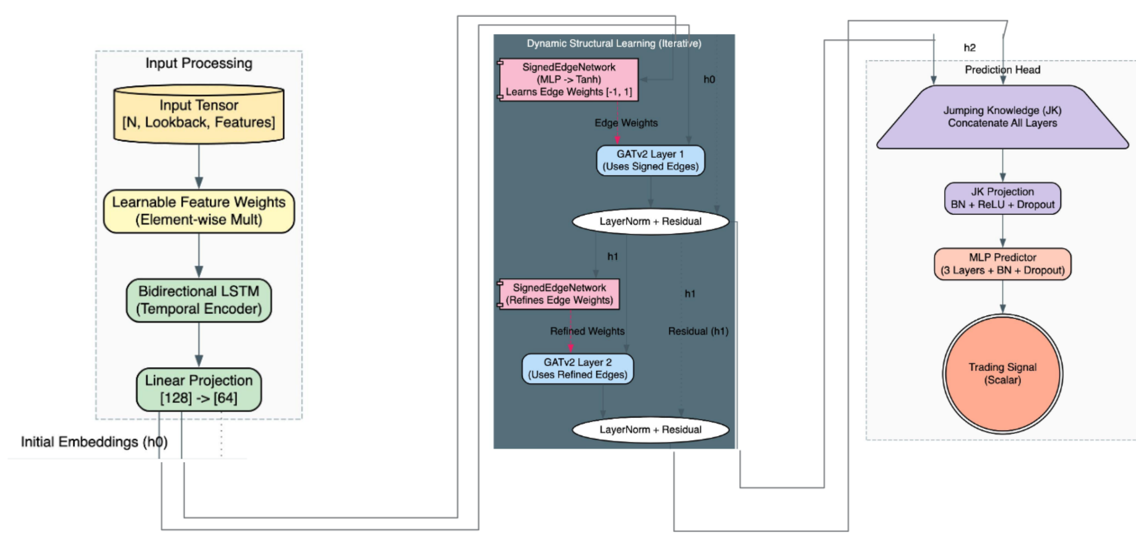

Figure 3: GNN model architecture

On each test day, we take the most recent 60-day window, normalize using train statistics, run the model, and obtain a score for each asset; we also log average feature-attention weights for diagnostics. Among tradable tickers (base ETFs only), we rank by score and form a long–short portfolio with equal weights within each sleeve by selecting the top group as longs and the bottom group as shorts (with sensible clipping when few names are available). Ratio series are never traded. Initial capital is split into H equal daily buckets; each day we close the bucket opened H days prior (realize P&L and return proceeds to cash) and open a new bucket following that day's ranks, producing staggered H-day holding periods that smooth timing noise. Daily total portfolio value equals invested value plus cash.

On each test day, we extract the most recent 60-day window, normalize features using *train-period statistics*, run the model, and obtain a score for each asset. We also log average feature-attention weights for diagnostics.

Among the tradable tickers (base ETFs only), we rank assets by score and form a long–short portfolio with equal weights within each sleeve: the top group forms the long side, and the bottom group forms the short side (with sensible clipping when fewer names are available). Ratio series are never traded.

Initial capital is divided into H equal daily buckets. Each day we close the bucket opened H days earlier (realizing P&L and returning proceeds to cash) and open a new bucket corresponding to that day's ranks, producing staggered H-day holding periods that smooth timing noise. Daily total portfolio value equals invested value plus cash.

4.3.3 Signal Pipeline and Portfolio Construction

We summarize the implementation details of the GNN strategy into three operational stages: (i) model optimization and ensemble training, (ii) signal generation with regime-aware adjustments, and (iii) portfolio construction and evaluation.

Model optimization and ensemble. We optimize each model with AdamW using learning rate 10^{-4} and weight decay 10^{-5}. Training uses a linear learning-rate warmup over 20 epochs, gradient clipping with maximum norm 2.0, and early stopping based on a rolling 10-epoch average of the validation Information Coefficient (Spearman correlation between predictions and realized returns) with patience 40 epochs. Mini-batches contain 32 calendar dates per batch. To reduce variance and improve robustness, we train an ensemble of $M = 3$ models with different random seeds (0, 43, 44). At inference time, we average predictions across ensemble members:

$$\hat{y}_i^{(ens)} = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_i^{(m)}.$$

Signal generation, filtering, and modifiers. Rather than using a static threshold, we implement *dynamic Z-score filtering* based on the rolling distribution of ensemble signals. We maintain a rolling buffer of the past 60 days of ensemble signals, compute the historical mean and standard deviation across all assets and time, and map current signals to Z-scores. We only trade when $|\text{z-scored signal}| > 1.5$. A minimum history requirement of 20 days is enforced before applying Z-score filtering; during warmup, a fallback threshold of 0.05 is used. Volatility indices (VIX, MOVE) serve as non-tradable signal modifiers that adjust position sizing based on market regime. We apply domain knowledge to amplify or dampen signals (e.g., if VIX is high $\rightarrow$ negative signal for equities, positive signal for safe havens). Ratio series (XLY/XLP, LQD/IEF, HYG/IEI, TIP/IEF) generate signals that are decomposed into their constituent legs: for example, a long signal on XLY/XLP translates to long XLY and short XLP. Leg signals are aggregated with base asset signals before position sizing.

Portfolio construction, backtesting, and metrics. Initial capital (1,000,000) is divided into $H = 5$ daily buckets of \$200,000 each. Each day we (i) close the bucket opened H days earlier to realize P&L and (ii) open a new bucket using the current day's signals, producing staggered H-day holding periods that smooth timing noise. Among tradable assets (base ETFs only; ratio series and modifiers excluded), we rank assets by the filtered ensemble signal, select the top n_{long} assets for long positions (30% of strong signals, minimum 1) and the bottom n_{short} assets for short positions (30% of strong signals, minimum 1). We equal-weight within each sleeve: $w_{\text{long}} = 0.5/n_{\text{long}}$ and $w_{\text{short}} = -0.5/n_{\text{short}}$. Strategy performance is evaluated using Sharpe ratio, maximum drawdown, win rate, and Information Coefficient. Benchmarks include an equal-weighted portfolio of base ETFs and SPY (S&P 500).

5 Results

We evaluate two strategies: a *Lévy lead–lag* baseline and a *GNN-driven* multi-asset strategy. Results below are preliminary and subject to further tuning and robustness checks (e.g., costs, capacity, and regime subsamples). *All numbers in this section are drawn from the current experiment log.*

5.1 Lévy Lead–Lag Strategy

Table 2 reports annualized performance for lookahead horizons $k \in \{1, 3, 5, 7, 10\}$ with lookback $W = 50$ and follower/leader percentile $p = 20\%$.

Table 2: Lévy long-leg performance across horizons k.

k	Ann. Return	Ann. Vol	Sharpe	MaxDD
1	0.1733	0.1960	0.8843	-0.2476
3	**0.2071**	**0.1748**	**1.1846**	-0.2248
5	0.1472	0.1911	0.7704	-0.2332
7	0.1835	0.2148	0.8543	-0.2263
10	0.1495	0.1662	0.8995	-0.2792

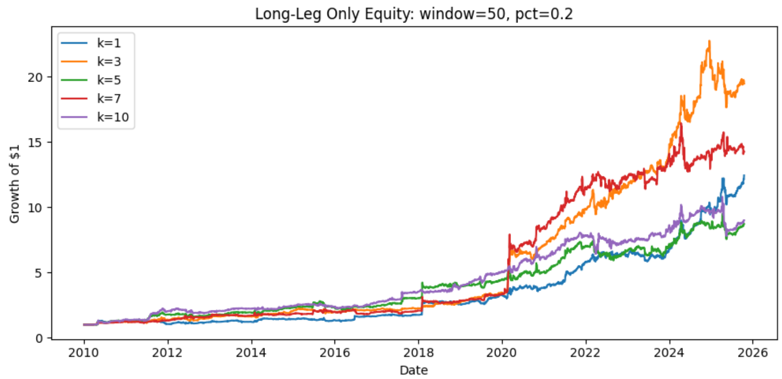

Figure 4: Cumulative equity curve of strategy vs benchmark over different holding periods

Figure 4 plots the cumulative equity of Strategy 1 for different holding periods k. We find that: (i) $k = 3$ currently shows the best risk-adjusted profile (Sharpe ≈ 1.18), with $\sim$21% annualized return at $\sim$17–18% vol and max drawdown $\sim$–22%. (ii) Other horizons are profitable but less efficient.

To assess predictive content, we examine follower excess returns in **Strategy 2** over an equal-weight market and cross-sectional information coefficients (IC). For brevity we show summary diagnostics in Table 3.

Table 3: Lévy diagnostics by k: follower excess vs. market and IC statistics.

k	Blocks (n)	Mean(F–M)	t-stat	Hit rate	Mean IC
1	2008	0.000525	1.51	0.509	0.0135
3	678	**0.002144**	**2.19**	0.519	**0.0331**
5	399	0.001629	1.06	0.536	0.0204
7	268	0.004669	1.54	0.500	-0.0025
10	208	0.003034	1.14	**0.572**	0.0396

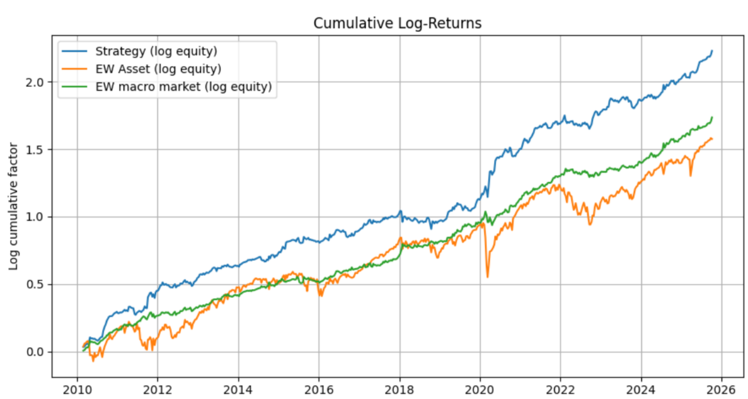

Figure 5: Cumulative log equity curve of strategy vs benchmark returns

Figure 5 plots the cumulative *log equity* $\log(V_t/V_0)$ of Strategy 2 alongside an equal-weight portfolio of tradable assets and an equal-weight macro multi-asset benchmark. Using log equity makes the growth path additive in time and facilitates comparison across strategies: vertical differences correspond to differences in long-run risk-adjusted performance rather than simply in dollar scale. The plot shows that the targeted SPY/EFA overlay delivers a consistently higher log wealth trajectory while largely sidestepping the sharp 2020 drawdown. This resilience arises because, during the COVID crash, the Lévy-based follower signal is rarely (if ever) triggered for SPY/EFA, so the strategy falls back to the diversified macro multi-asset portfolio instead of taking concentrated equity risk.

5.2 GNN-Driven Strategy

We train a temporal, multi-relational GNN to forecast next-period returns per ETF and convert scores into a risk-controlled long–short portfolio.

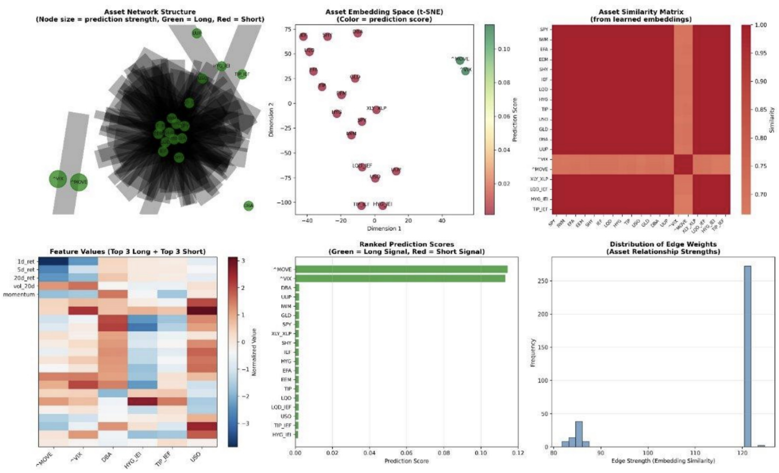

Figure 6: GNN-Driven Strategy: Performance (early stages)

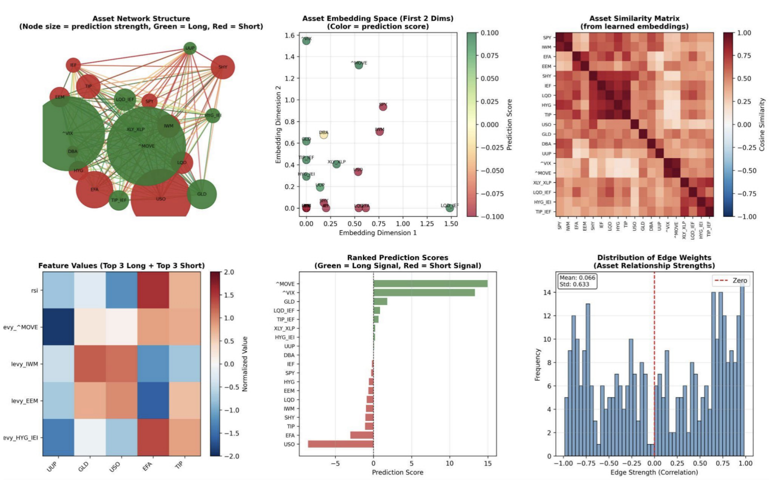

Figure 7: GNN-Driven Strategy: Performance

5.2.1 Model Diagnostics and Signal Interpretation

We summarize the GNN's learned market structure and the resulting cross-sectional signals along four complementary views: embeddings, signed connectivity, ranked exposures, and feature attributions.

Learned asset structure (embeddings and similarity). The 2D asset embeddings reveal economically intuitive clustering without any class supervision. Volatility and defensive proxies (MOVE, VIX) group tightly, suggesting the model treats them as a coherent risk-regime anchor. Core equity and credit instruments (SPY, HYG, LQD) sit centrally, while ratio/spread indicators (LQD_IEF, HYG_IEI, XLY_XLP) form a distinct subcluster consistent with their role as relative-value state variables. Commodities (GLD, USO, DBA) appear more isolated, indicating weaker or more episodic integration with the rest of the macro network. This interpretation is reinforced by the asset similarity matrix, which shows strong within-block structure for equities (SPY/IWM/EFA/EEM), a rates/credit grouping (LQD/HYG/IEF), and a separate ratios cluster (XLY_XLP, LQD_IEF, HYG_IEI, TIP_IEF). As training progresses, the similarity map sharpens and begins to exhibit clear anti-correlations—most notably equities versus defensives such as SHY and GLD—suggesting that the model internalizes a risk-on/risk-off partition rather than merely static correlation.

Signed edge dynamics. The learned signed edge weights are centered near zero with wide dispersion and mild bimodality, indicating a mix of positive and negative ties with heterogeneous strength across the universe. Table 4 summarizes the distribution. Qualitatively, the early-stage graph appears dense and radial with many weak links, while the refined graph evolves toward a clearer hub-and-spoke topology in which volatility nodes act as dominant hubs. This is consistent with a market structure where regime-sensitive assets mediate information flow across risk assets, rates, and defensive exposures.

Table 4: Signed edge-weight summary.

	Mean	Std
Weights	0.066	0.633

Signal profile and selectivity. Ranking by the model's cross-sectional scores yields a defensive tilt. The strongest long candidates are MOVE, VIX, GLD, and LQD_IEF (scores ≈ 0.02–0.12), while the strongest shorts include USO, EFA, EEM, and HYG_IEI. The asymmetry in score strength (shorts stronger than longs) is consistent with a risk-off stance in the evaluated period. Applying dynamic filtering with $|Z| \geq 1.5\sigma$ removes roughly 40–60% of low-conviction names, highlighting a deliberate *quality-over-quantity* design in execution; the resulting win rate is 43.28%.

Feature-level attribution. Examining feature contributions offers a coherent behavioral narrative: top-long signals are typically driven by Lévy-area interactions with high-volatility assets, negative RSI (mean reversion), and mixed short/medium horizon momentum (1d/5d/20d). By contrast, top-short signals tend to coincide with positive Lévy exposure to risk assets, extended 20-day rallies, and low realized volatility relative to history (suggesting complacency). Taken together, these patterns suggest the model combines mean-reversion dynamics with directional lead–lag information to produce contrarian tilts that may be particularly effective in range-bound or transition regimes.

Figure 8: GNN-Driven Strategy: PnL

Table 5 compares the current *Enhanced GNN Ensemble* against an equally weighted (EW) benchmark and SPY buy-and-hold.

Findings: (i) The GNN surpasses EW on both absolute and risk-adjusted metrics (+13.97pp total return; Sharpe 1.86 vs. 1.69) with comparable drawdown control. (ii) Versus SPY, the GNN runs a lower-volatility profile with a *higher* Sharpe (1.86 vs. 1.77) and substantially smaller drawdown (–5.5% vs. –10.0%), albeit with lower total return in a strong equity tape—consistent with a diversified long–short design.

Summary. The Enhanced GNN Ensemble strategy successfully validates the graph neural network approach for multi-asset trading. The strategy's ability to adapt graph structure dynamically (shown in edge weight evolution) and integrate macro

Table 5: GNN strategy vs. benchmarks (preliminary).

Strategy	Total Ret. (%)	Ann. Ret.	Sharpe	MaxDD
Enhanced GNN Ensemble	30.86	0.1993	**1.86**	-0.0554
Equal-Weighted Benchmark	16.89	0.1112	1.69	-0.0464
SPY (Buy-and-Hold)	36.09	0.2315	1.77	-0.0997

context (VIX/MOVE modulation) demonstrates that GNNs provide a powerful framework for financial prediction that outperforms traditional approaches. While there is room for improvement in trending markets, the overall results strongly support continued development of graph-based trading systems.

6 Conclusion

This project reframes macro markets as an evolving network and uses a spatio–temporal Graph Neural Network (GNN) to turn that structure into tradable signals. By representing exchange–traded funds as nodes and their time–varying dependencies (correlation, lead–lag, spreads) as edges, the model forms context–aware expectations for each asset and issues cross–sectional return scores that are converted into a risk-controlled long–short portfolio. This network view addresses a central limitation of single-series or static-covariance methods: relationships across assets change with regimes, and useful information often propagates through the graph before it appears in any one series.

Results indicate that the GNN produces competitive, after-volatility performance with materially lower drawdowns than simple benchmarks, while a Lévy lead–lag baseline provides corroborating timing evidence. Qualitative diagnostics, like 2D embeddings, similarity matrices, and signed edge distributions, suggest the model learns economically coherent structure (e.g., defensives and volatility hubs, equity/credit blocks, and ratio clusters) without supervision on asset classes. Together, these findings support the feasibility of graph-based signals for regime-aware tilts and hedge alignment in a diversified ETF universe.

These conclusions are necessarily tentative. Backtests remain sensitive to architecture and hyperparameters; transaction-cost and capacity modeling are simplified; and durability must be demonstrated across longer histories and stress windows. Governance requirements, such as versioned data, leak-free evaluation, and attribution of decisions, remain an active focus. Going forward, we will expand the sample, harden the cost and turnover model, and run comprehensive ablations (no-graph vs. graph; undirected vs. directed edges; with/without macro context). A shadow, out-of-sample run will test live stability.

We also see several natural extensions to the Lévy-based baselines. First, we will explore a *long–short* Lévy implementation that explicitly pairs follower longs with a short leg (e.g., short leaders or short the weakest expected assets) to better isolate lead–lag alpha from broad market drift, while carefully stress-testing drawdown asymmetry and turnover under realistic cost assumptions. Second, rather than defining leaders and followers purely by percentile ranks of node-level scores, we will investigate *cluster-level* lead–lag structure using directed-network community detection. In particular, Bennett et al. propose detecting statistically significant leading and lagging groups in financial markets via directed clustering methods, including *Hermitian clustering* that operates on the complex Hermitian transformation $\tilde{A} = i(A - A^{\mathsf{T}})$ of a directed adjacency matrix and its random-walk-normalized variant (Hermitian RW), which is designed to recover communities with strong flow imbalance. [6] Adapting this idea to our macro ETF Lévy networks could yield more stable, economically interpretable leader–follower *clusters* and enable cluster-rotation or cluster-spread portfolios as a structured alternative to single-asset ranking.

With continued validation, the combined framework—rule-based Lévy lead–lag tests plus nonlinear spatio–temporal GNN learning—offers a practical, low-turnover overlay for risk management and tactical allocation in macro portfolios.

References

[1] Xingyue Pu, Stephen Roberts, Xiaowen Dong, and Stefan Zohren. Network momentum across asset classes, 2023. Working paper.

[2] Linze Li and William Ferreira. Follow the leader: Enhancing systematic trend-following using network momentum, 2025. Working paper.

[3] Seong-Min Yoon, Md Al Mamun, Gazi Salah Uddin, and Sang Hoon Kang. Network connectedness and net spillover between financial and commodity markets. *The North American Journal of Economics and Finance*, 48:801–818, 2019.

[4] Álvaro Cartea, Mihai Cucuringu, and Qi Jin. Detecting lead-lag relationships in stock returns and portfolio strategies. *SSRN Electronic Journal*, November 2023.

[5] Ali Rahimi. Portfolio construction using lead–lag relationships discovery. `https://github.com/ARahimiQuant/lead-lag-portfolios`, 2023. GitHub repository.

[6] Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. Lead–lag detection and network clustering for multivariate time series with an application to the us equity market. *Machine Learning*, 111(12):4497–4538, 2022.