



Zero-Copy Semantic Contagion

An In-Memory Streaming Architecture for Evolving Attention Graphs

Kabir Murjani
Nirma University

FinDS: Workshop on Data Management for
the Modern Financial Systems

May 31, 2026 | Bengaluru, India

The ~~blind~~ spot in per-ticker forecasting

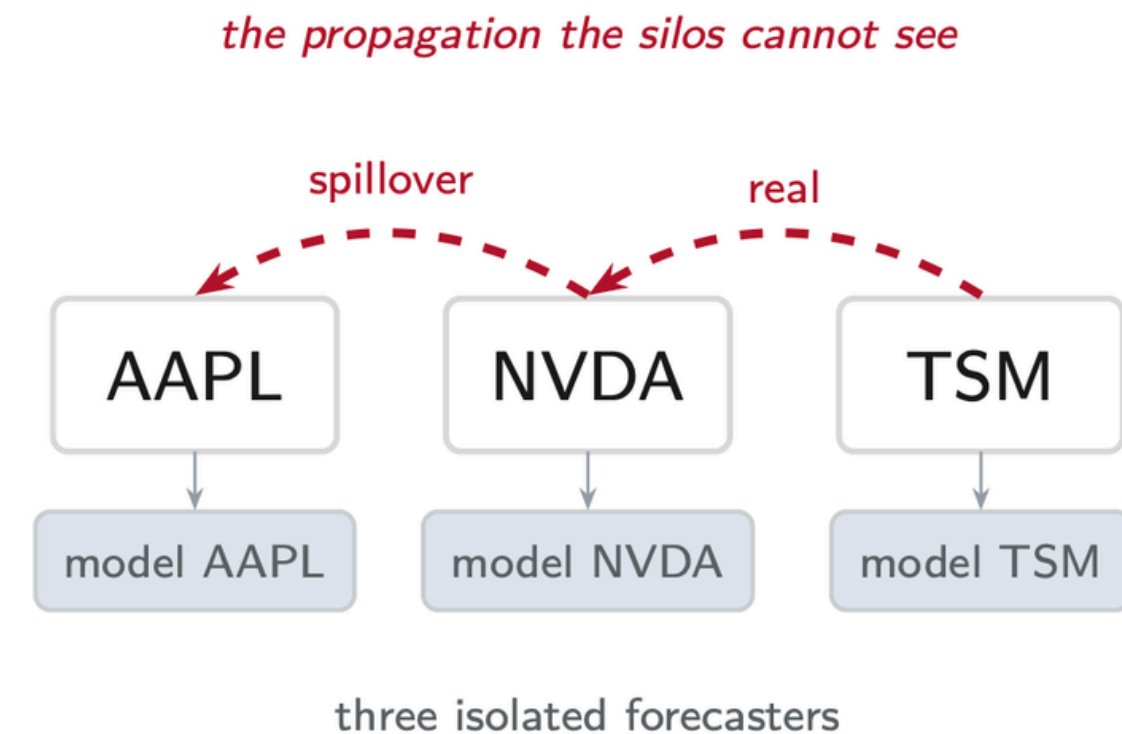
Production forecasters model one company at a time. Each equity is conditioned only on:

- its own price history, and
- news tagged with its own ticker

Apple's price model is conditioned on Apple's own history and Apple-tagged news. A foundry disruption in Taiwan cannot reach it until either Apple's price has already moved; or a journalist writes the word "Apple."

So cross-company shocks, supply-chain links, shared narratives, joint volatility, are invisible by construction **until the price has already moved.**

Background



Returns propagate along supply-chain links [Cohen & Frazzini '08] and volatility channels [Diebold & Yilmaz '14] but both read realised returns, after the fact.

Two barriers

Barrier 1 — Latency

Off-the-shelf semantic extractors add **hundreds of ms**. The price has moved before inference even begins.

Our answer. Push parsing, time stamping, ticker scanning, and de-duplication into a zero-allocation Rust edge: $\sim 2.1 \mu\text{s}$ per record.

Barrier 2 — Representation

A static graph is a poor model of attention, which **decays between events** and demands explicit continuous time.

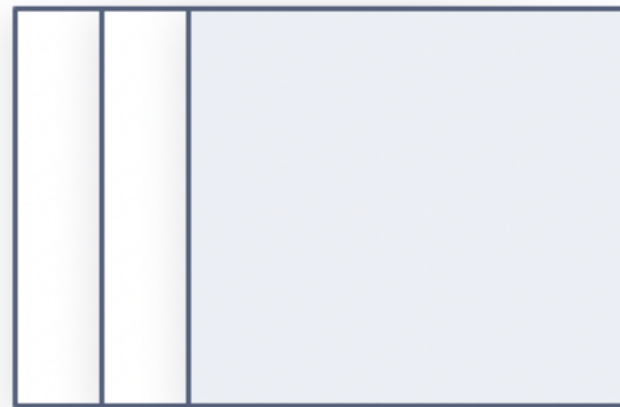
Our answer. A multivariate Neural Hawkes Process: per-node hidden states that decay between events and jump on arrival.

Our heterogeneous **Rust / Python** architecture maps cross-company attention as a **continuous-time graph driven directly from text**.

Continuous-time graph learning, not snapshots

Attention between companies decays rapidly between events. Representing it faithfully requires explicit continuous-time dynamics not a sequence of static graph snapshots stitched together at fixed intervals.

Snapshot / Discrete Graphs



frozen windows

- Adjacency fixed within each window
- Temporal resolution capped by window size
- Decay between events is not modeled

Our Continuous Time View

$$\alpha \cdot e^{-\delta(t-t_k)}$$

- Each node state evolves in continuous time
- Excitation decays smoothly between events
- Edges scored at the instant of each event

Built on the Neural Hawkes process (Mei & Eisner, 2017) on a learned, self-modulating point process.

Contributions

1 Zero-copy Rust ingestion edge

Parse (~ 100 ns), scan 47 tickers (~ 1.2 μ s), monotonic stamps (~ 25 ns).

2 Multivariate Neural Hawkes core.

Per-node continuous-time LSTM + bilinear latent projection $\Rightarrow$ directed, asymmetric excitation.

3 Adaptive edge-pruning rule.

Bounded graph density under perpetual streaming.

4 A contagion-detection protocol.

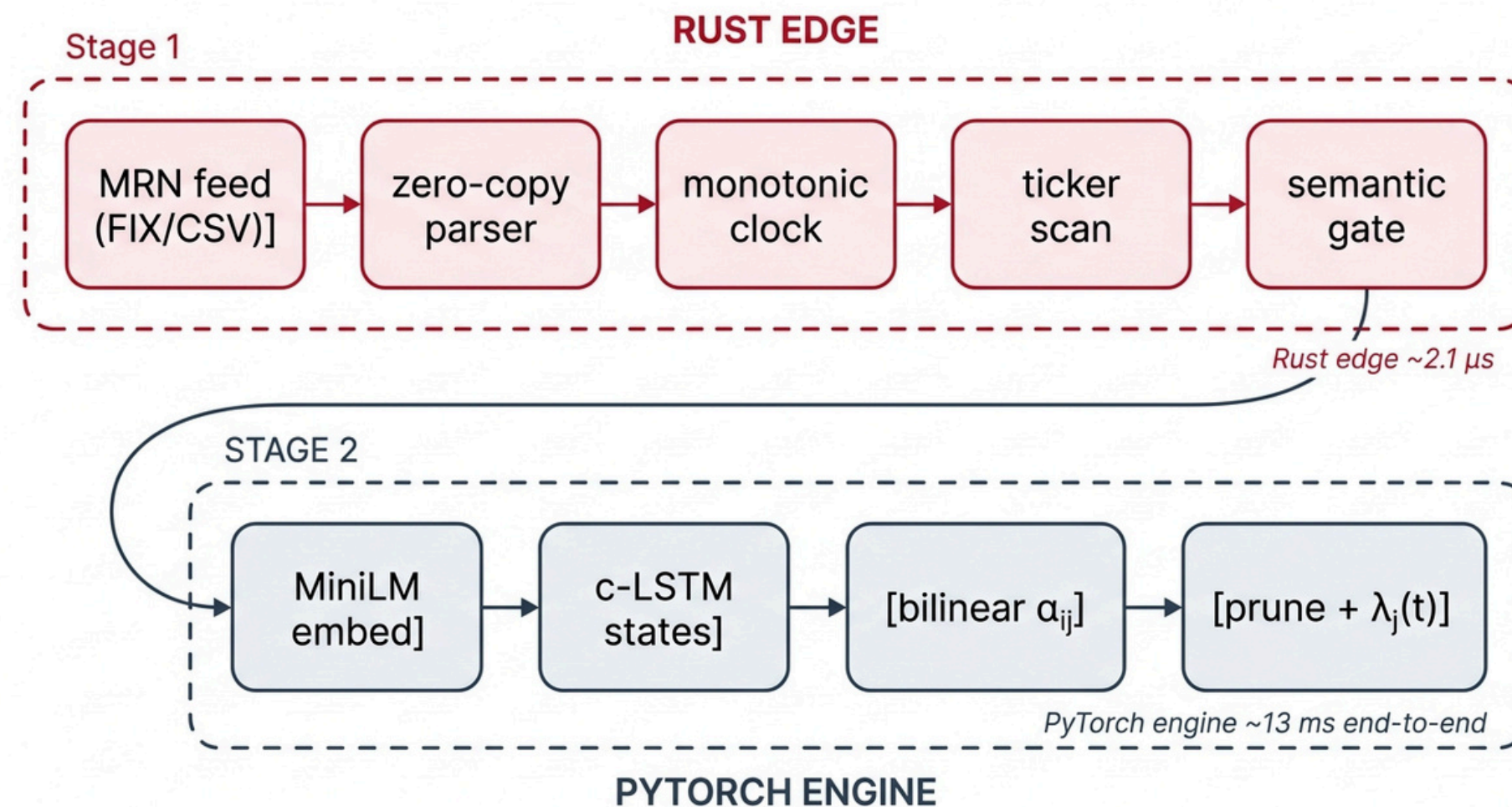
Strict temporal holdout; 1.70 $\times$ lift over random, 3.36 $\times$ over a sector heuristic at the 90th-pct threshold.

Complementary to per-ticker forecasting: given a shock to one firm, which others move next and when?

Architecture

Key Design Decisions

1. **Zero-copy parsing:** Byte-slice references into original input buffer. No heap allocation, preserving cache locality under burst traffic.
2. **Frozen sentence embeddings:** MiniLM-L6-v2 (384-d, 22M params). ~8 ms per article on **CPU**.
3. **Monotonic timestamps:** Enforced event ordering. RDTSC-ready on x86-64 for sub-nanosecond resolution.
4. **Process boundary separation:** Rust ingestion decoupled from PyTorch inference. Math invariant to deployment target.

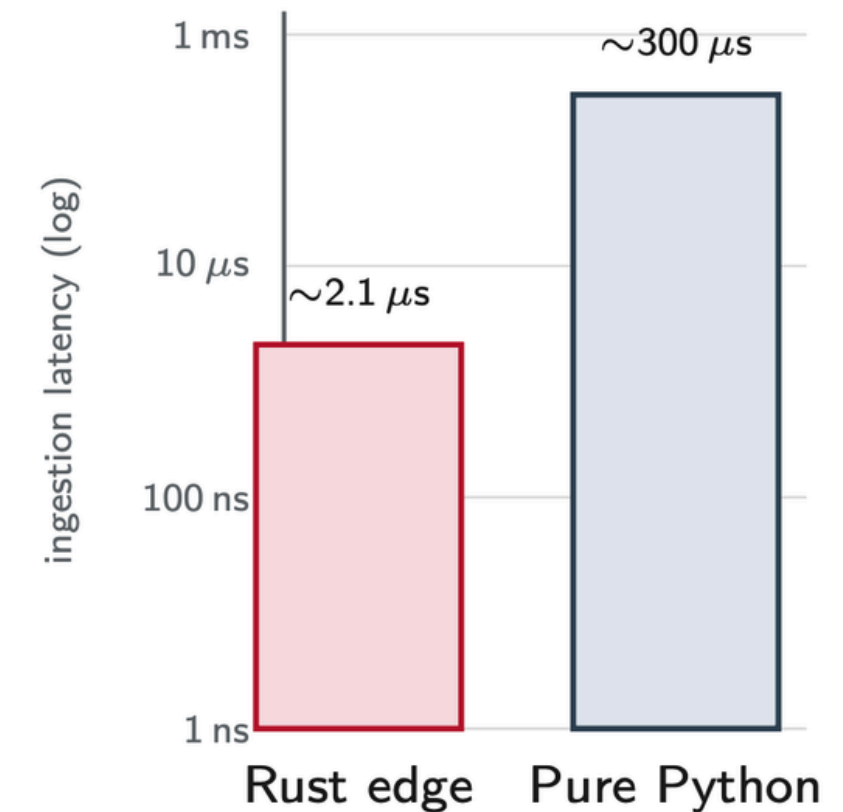


Why Rust? Why Zero Copy?

Zero-copy. The parser returns **byte-slice references into the original buffer** which no heap allocation on the hot path. This avoids the memory churn that destroys cache locality under burst traffic, and eliminates GC tail latency entirely.

The ablation that matters.

Swap the edge for equivalent pure Python: ingestion latency rises **~2 orders of magnitude**, with GC pushing P99 into the millisecond range that for bitwise-identical model output.



> 400 K records/sec on a single ARM core.

Model

The continuous-time engine

Conditional Intensity

Learned excitation in continuous time

Classical Hawkes [Hawkes '71] with the parametric kernel replaced by a learned, context-dependent α .

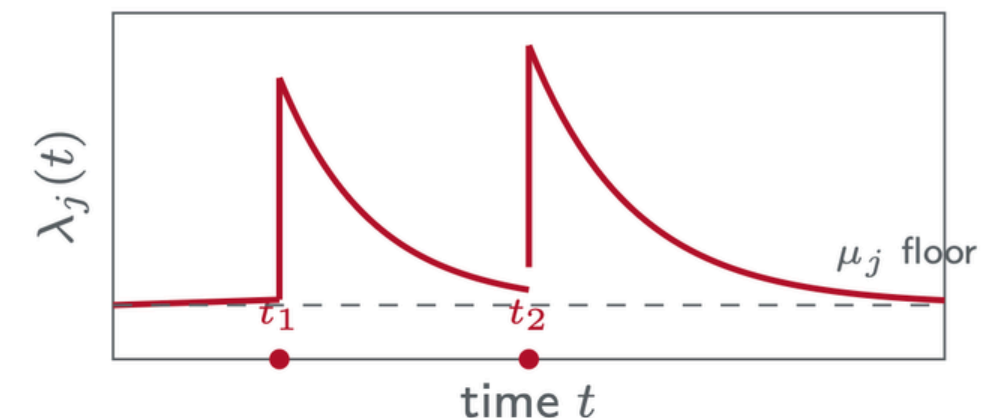
$$\lambda_j(t) = \phi\left(\mu_j + \sum_{i \in \mathcal{N}_t(j)} \alpha_{ij}(t_{k_i}) e^{-\delta_j(t-t_{k_i})}\right)$$

$\phi = \text{softplus}$ keeps intensity positive.

$\mu_j > 0$: ticker j 's resting attention level.

$\delta_j > 0$: learned decay — attention **fades** between events.

$\mathcal{N}_t(j)$: the *dynamic* neighbourhood at time t .



jump on arrival, exponential decay between

Continuous-time LSTM

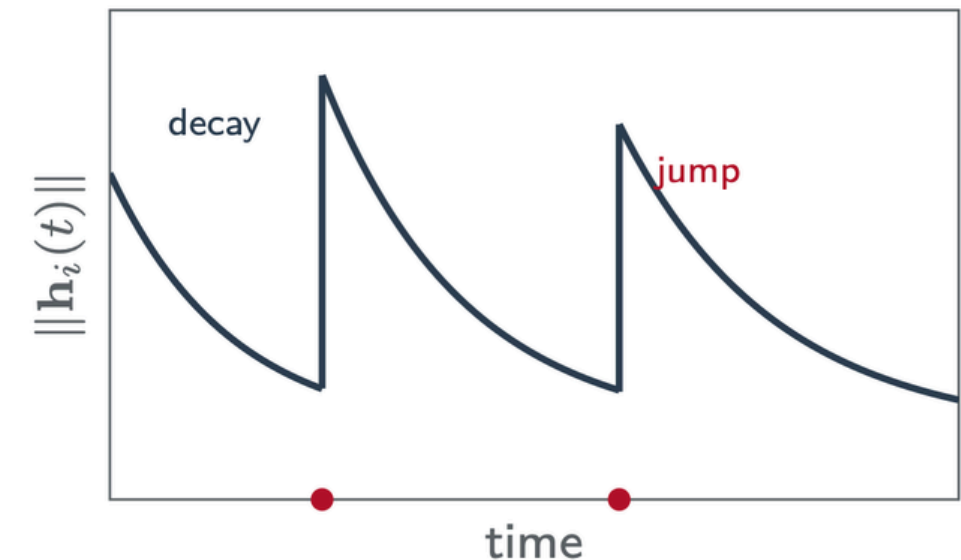
A hidden state that lives between events

Why a c-LSTM [Mei & Eisner '17]? A vanilla RNN only updates at events. Attention, though, evolves continuously, it must decay in the silence between headlines.

Between events the cell drifts toward a target $\bar{\mathbf{c}}_i$:

$$\mathbf{c}_i(t) = \bar{\mathbf{c}}_i + (\mathbf{c}_i(t_k) - \bar{\mathbf{c}}_i) \odot e^{-\gamma_i(t-t_k)}$$

At arrival a standard LSTM gate update fires on the projected embedding $\tilde{\mathbf{v}}_k = W_e \mathbf{v}_k$.



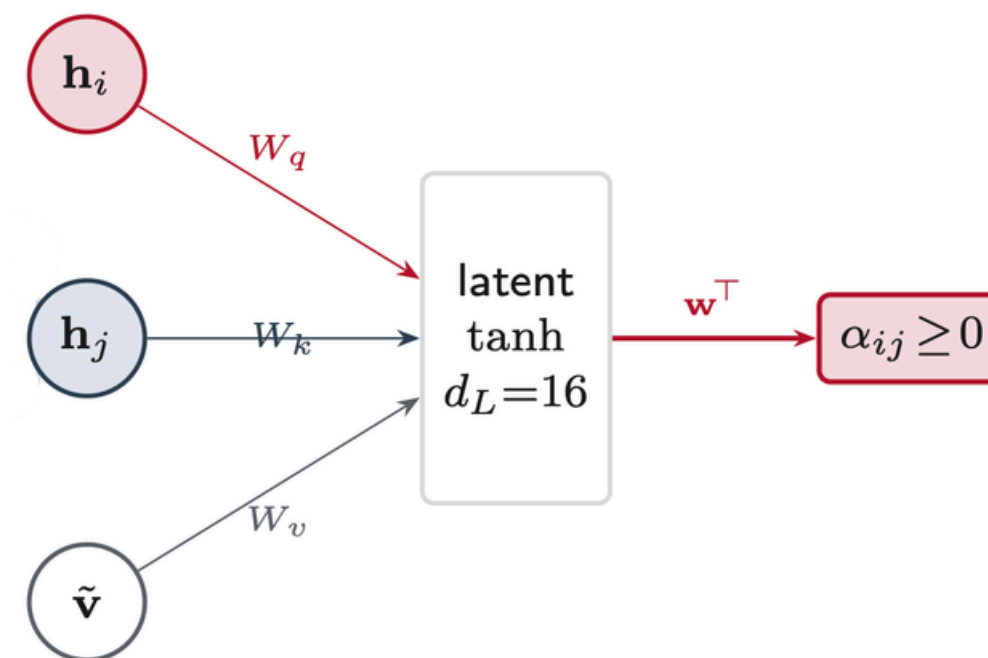
Each of 47 nodes carries its own such state.

Bilinear Latent Projection

making $\alpha_{ij} \neq \alpha_{ji}$ possible

Directed excitation is scored by a bilinear interaction [Kim et al. '18] of source state, target state, and the event embedding:

$$\alpha_{ij} = \text{softplus}(\mathbf{w}^\top \tanh(W_q \mathbf{h}_i + W_k \mathbf{h}_j + W_v \tilde{\mathbf{v}}))$$



Because source and target enter through *different* projections W_q, W_k , the excitation is **inherently asymmetric** — $i \rightarrow j$ need not equal $j \rightarrow i$.

A symmetric construction (correlation, or concatenation attention) has asymmetry index 0.

Adaptive Edge Pruning

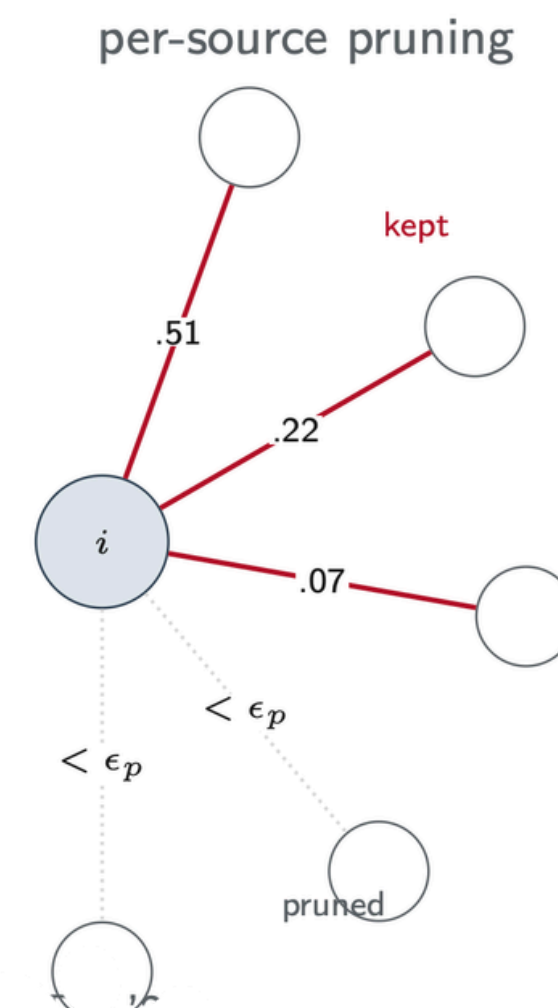
Bounded density under perpetual streaming

An infinite stream grows the graph without bound. We prune by a **cumulative excitation score**:

$$S_{ij}(t) = \sum_k \alpha_{ij}(t_k) e^{-\delta_j(t-t_k)}$$

Edges with $S_{ij} < \epsilon_p$ are dropped (with a minimum-neighbor floor).

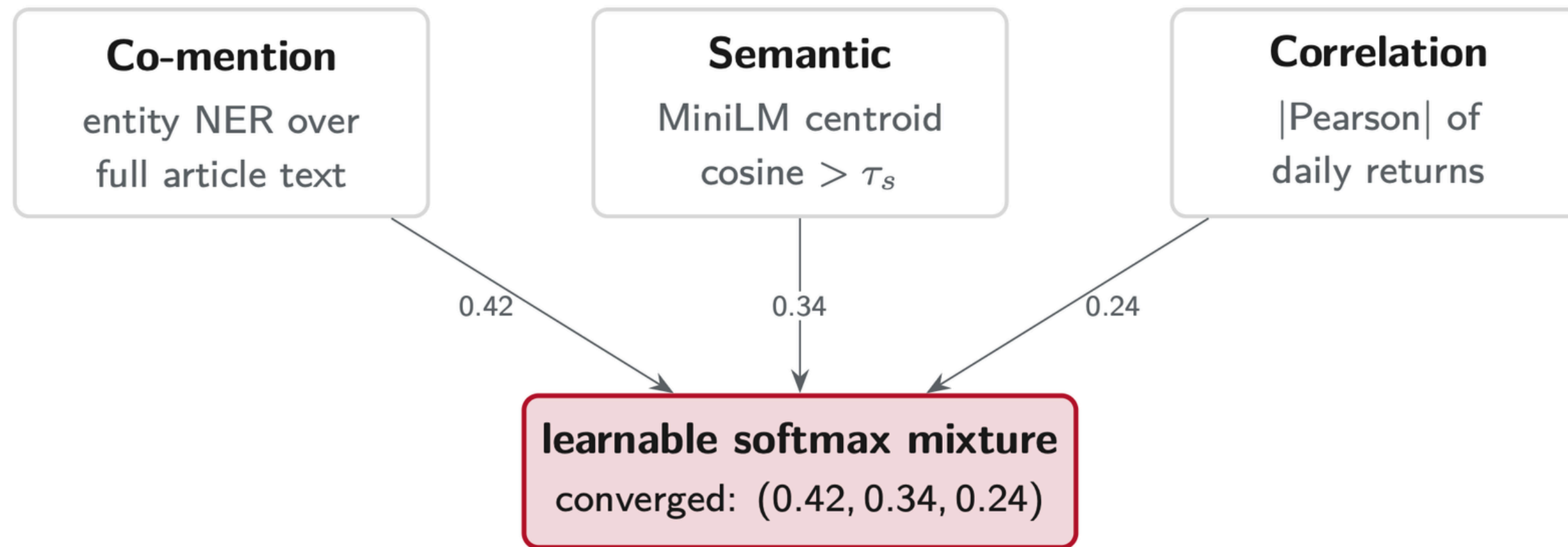
Proposition 1. In-degree is bounded:
 $d_j(t) \leq \lfloor \lambda_{\max}/\epsilon_p \rfloor$, independent of
stream length.



Graph Construction

A learned mixture of three real signals

Weights are co-optimised with the model. The prior co-mention is the strongest linkage signal is preserved after training, and the AAPL→NVDA→TSM supply chain emerges **without any explicit annotation.**



Dataset

FNSPID, sliced to July 2022

FNSPID, Dong, Fan & Peng, KDD 2024 a comprehensive financial-news time-series corpus pairing timestamped headlines with OHLCV and GPT-scaled sentiment.

THE FULL CORPUS

29.7 M

daily price records

15.7 M

time-stamped news articles

4,775

S&P 500 companies

1999–2023

temporal coverage

We restrict to **July 2022**: a month of elevated cross-sector volatility which is semiconductor supply constraints, Fed rate decisions, EV-battery shortages.

The engine uses only raw text embeddings (MiniLM-L6-v2); sentiment is consumed solely by the per-ticker baselines.

Results

Does the graph actually carry signal?

Contagion detection

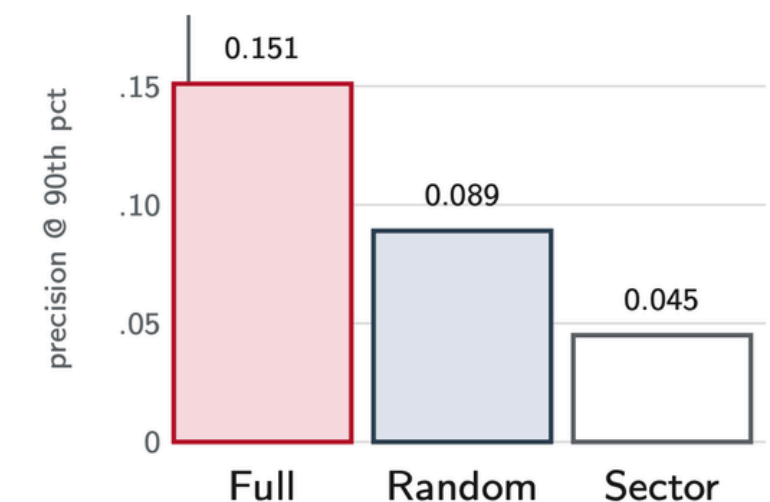
Top-3 targets per source event; a hit = target's next-day $|return|$ exceeds its pre-holdout percentile.
Thresholds estimated on pre-holdout data only.

Lift stays in the $1.36\text{--}1.81\times$ band across all five thresholds that's not a single cherry-picked operating point.

Three structural ablations. One moves **everything**.

Removing the adjacency collapses precision to **zero** at every threshold. The graph topology is the sole mechanism of cross-company signal and the c-LSTM dynamics.

Precision @	75	80	85	90	95
Full model	.281	.226	.165	.151	.095
Random	.178	.125	.109	.089	.069
Sector	–	–	–	.045	–
Lift vs. random	1.58	1.81	1.51	1.70	1.36



$1.70\times$ over random, **$3.36\times$** over sector.



Case Study

The semiconductor chain

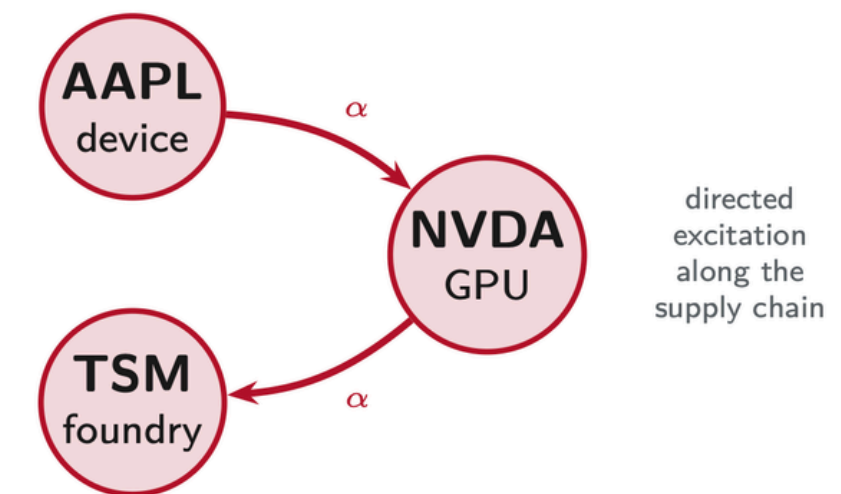
A real shock down the chip chain

1 July, 2022

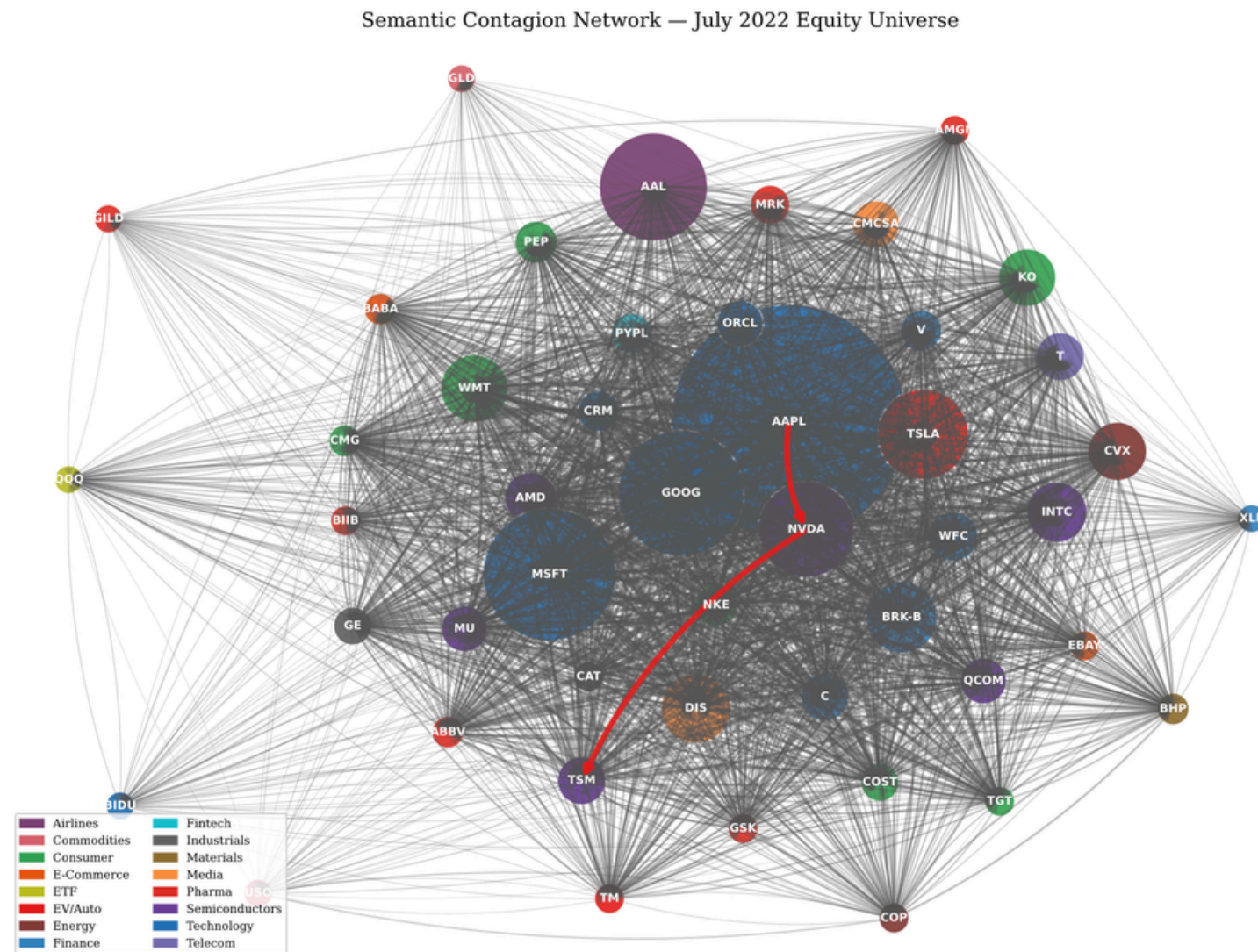
Taiwan Electronic Times reports that TSMC's three largest customers on Apple, AMD and Nvidia that all cut order targets; **Nvidia asked to delay and reduce Q1 orders** as terminal demand collapsed. TSMC rarely sees three majors adjust at once.

The model's learned path is exactly this supply chain. A single foundry-demand shock propagates across structurally connected names directed excitation spikes with AAPL news and decays via the c-LSTM dynamics.

This directed channel tracks the aggregate envelope at $\sim 1/10$ magnitude (out-degree 0.96).



Learned contagion network of July 2022



Reading the graph

- 47 nodes: node size $\propto$ article mention count.
- Edge weight $\propto$ the combined adjacency A_{ij} .
- Densest excitation concentrates around semiconductor and technology names.

Highlighted: the AAPL $\rightarrow$ NVDA $\rightarrow$ TSM semiconductor chain — recovered from co-mention frequency and semantic similarity **alone**, with no explicit supply-chain annotation.

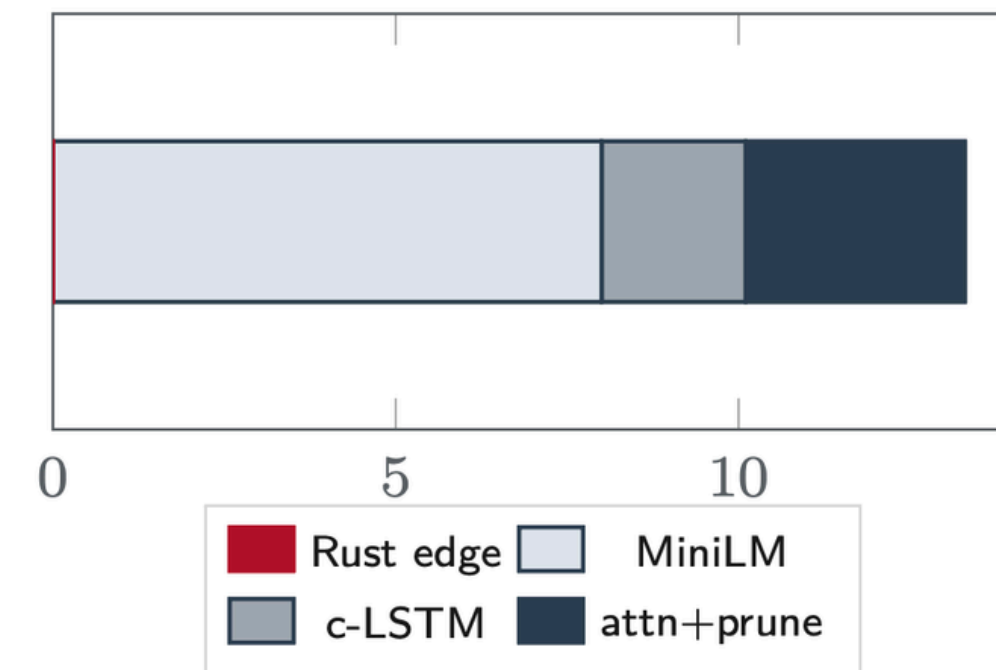
Figure 2. Contagion network, July 2022 (47 nodes). Highlighted: Apple $\rightarrow$ NVIDIA $\rightarrow$ TSMC semiconductor chain.

Latency Analysis

The edge is free but the bottleneck is the encoder.

The Rust edge is <0.02% of end-to-end latency. No GPU. The encoder dominates and INT8 or distillation reduces it.

Stage	Latency
CSV zero-copy parse	102 ns
Monotonic timestamp	25 ns
Ticker scan (47)	1.2 μ s
Cosine (384-d)	317 ns
Semantic gate	507 ns
Rust edge total	$\sim 2.1 \mu$s
MiniLM embedding	~ 8 ms
c-LSTM update (47)	~ 2.1 ms
Bilinear + prune	~ 3.2 ms
End-to-end	~ 13 ms



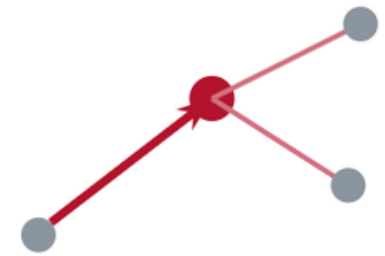
Outlook

Limitations

- **Tiny, single-month corpus.** 638 articles / 47 tickers over July 2022.
- **Daily-resolution evaluation, not intraday.** The microsecond/millisecond architecture is validated, but every contagion metric is computed at next-trading-day granularity (FNSPID only exposes daily OHLCV), so the "61.5h lead" is an artifact of the tape.
- **Modest absolute precision.** 15.1% at the 90th percentile in the result is a lift over baselines (1.70× random, 3.36× sector), not a high-accuracy detector; usefulness rests on the relative comparison, not standalone performance.
- **Deployment claims are projected, not measured.** Shared-memory IPC, RDTSC timestamping, AVX-512 scanning, kernel-bypass networking, and the FIX-feed integration are described as future/estimated; the prototype runs single-process on one CPU with a frozen MiniLM encoder as the real bottleneck.

Future Work

- **Intraday-resolution evaluation:** Sync an **institutional intraday tape** (TAQ minute bars) to an MRN feed, to test whether the 61.5 h lead compresses toward the 20–60 min microstructure regime.
- **End-to-end embedding tuning:** Jointly train the sentence encoder with the **Hawkes process via LoRA**, specializing the representation for contagion.
- **Higher-order propagation:** Stack Hawkes layers for **multi-hop cascades** ($i \rightarrow j \rightarrow k$), capturing deeper supply-chain effects.
- **Expanded temporal & instrument scope:** **Multi-month corpora** (full **FNSPID**, **GDELT**); beyond equities to any correlated-instrument market.
- **Attention-based graph constructor:** Let the topology adapt at each event from c-LSTM states, replacing the fixed convex mixture.



Thank you!

Kabir Murjani

kabir.murjani@iimb.ac.in, 23bee064@nirmauni.ac.in

Code and data: github.com/kcbir/zcsc