Event-Type Propagation Graphs: A Lag-Aware Layer for Predictive Financial Analysis

Yuxuan Li[1,*], Bohan Zhang[1,*], Guo Yu[1], Ye Gao[1], Haipeng Zhang[1,†], Yixi Zhou[1,2]
[1]ShanghaiTech University [2]Hong Kong Baptist University
{liyx2023,zhangbh2023,yuguo2022,gaoye2022,zhanghp,zhouyx2022}@shanghaitech.edu.cn

***Abstract*—Financial news affects stock movements through delayed event pathways: an inflation report may later shape interest-rate expectations, while litigation news may precede governance changes or analyst reassessment. Existing news-driven predictors usually compress each item into sentiment, text embeddings, or stock-level relations, leaving no reusable representation of which financial event types tend to precede later event- or price-relevant signals and at what trading-day delays. We propose CausalLagStock, a lag-aware event-type propagation framework for predictive financial analysis. It grounds raw news into structured event records with event type, affected ticker, event-impact magnitude, surprise, and novelty evidence; learns a sparse global event-type propagation graph whose directed edges encode lagged predictive associations and preferred delays; and injects the learned graph into downstream predictors, including a graph-enhanced Neural ODE. Experiments on two aligned financial-news stock movement datasets show improved accuracy and weighted-F1 under chronological evaluation. The learned graph also exposes delayed pathways such as inflation-related news preceding interest-rate signals and litigation-related news preceding governance changes, making it useful as an inspectable monitoring layer while balanced prediction of threshold-sensitive minority classes remains challenging.**

***Index Terms*—financial data mining, stock movement prediction, event extraction, lag-aware event propagation, large language models**

I. Introduction

Stock movement prediction supports applied financial decision making, including risk monitoring, analyst review, and horizon-specific stress testing. Financial news is a natural information source for this task, but a headline rarely acts as an isolated one-day signal. It first reports an *event type*, such as earnings, inflation data, litigation, merger activity, analyst revision, or executive change. Different event types become useful at different horizons: an earnings surprise may be absorbed quickly, while a lawsuit may be followed by analyst reassessment, compliance disclosure, or executive turnover over several trading days. We use *event propagation* to describe these temporally ordered predictive pathways among event types, not to claim that one news item interventionally causes another.

This distinction matters because similar headlines can carry different predictive value depending on market context. A reported number is informative only relative to expectations, and repeated coverage is less informative than a new disclosure. A useful news-driven framework should therefore identify the event type, ground it with evidence such as surprise and novelty, and represent when that type tends to become useful for later event- or price-relevant signals.

Existing news-driven predictors leave this representation problem only partially addressed. Many models compress a news item into a sentiment score or latent text embedding [1], [2], while event-driven variants use structured event features as static inputs [3]. These approaches can improve prediction, but they provide limited visibility into the event chain an analyst would inspect. In parallel, lag-aware financial methods have been developed to recover directional temporal dependence among signals, returns, and stocks, providing structured representations of how information and market effects evolve across time [4], [5]. Such structures are valuable, but the unit of analysis is usually an asset or time series rather than an event type. A stock-to-stock edge may show that two assets are temporally related; it does not explain whether the relation is driven by litigation, executive turnover, macro data, analyst revision, or another financial event category.

This paper argues that the missing abstraction is a reusable event-type propagation layer: a graph whose nodes are financial event types and whose directed edges summarize which event types tend to precede later event- or price-relevant signals, together with their preferred trading-day delays. A learned edge from inflation data to interest-rate events, for example, means that inflation news has repeatedly provided useful information for later rate-related signals around a preferred delay. Rather than treating a stronger text encoder as the main solution, we learn this global graph as an intermediate layer between raw news and downstream prediction. The graph is not a claim of interventionally identified causality from observational news data. Instead, it is a compact, inspectable, and reusable representation of lagged predictive associations under chronological supervision.

We instantiate this idea in CausalLagStock, an applied event-propagation mining framework for financial news-driven stock prediction. The framework converts raw news into event records, learns a sparse lag-aware event-type propagation graph, and evaluates that graph in forward-in-time prediction and causal interpretation. The goal is not only to improve a benchmark measure, but also to produce an event propagation layer that can be inspected in analyst-facing workflows such as risk monitoring and horizon-aware stress testing.

*Equal contribution. †Corresponding author.

We summarize the contributions of our paper as follows:

- **Lag-aware event-type propagation.** We formulate event-type propagation as an intermediate abstraction for financial news-driven prediction, shifting delayed temporal structure from isolated text instances or stock-level relations to reusable financial event types.
- **Grounded propagation mining framework.** We propose CausalLagStock, which grounds news with schema-constrained event extraction, surprise, novelty, and event-impact magnitude evidence, then learns a sparse global propagation graph with directed strength and preferred trading-day delay.
- **Reusable predictive and analytical layer.** We inject the graph into chronological stock movement predictors and analyze it as an inspectable monitoring layer. Experiments on FNSPID and CMIN show gains in accuracy and weighted-F1, expose delayed pathways for risk monitoring, and reveal the remaining difficulty of balanced Flat-class prediction. The implementation of CausalLagStock is available at GitHub [1].

II. Related Work

A. Financial Text Signals and Stock Forecasting Benchmarks

News-driven stock prediction has long used textual signals to complement price history. Early neural systems showed that tweets, financial news, and structured events can improve movement prediction when combined with market time series [1]–[3]. Recent work has shifted the setting toward larger benchmarks and stronger market-wide predictors. FNSPID provides a large-scale aligned corpus of stock prices and financial news and shows that dataset scale and sentiment information affect forecasting performance [6]. At the model level, recent stock forecasting architectures such as MASTER and StockMixer focus on cross-stock and cross-time price dependencies, while MDGNN models dynamic multi-relational stock graphs [7]–[9]. These methods strengthen temporal and relational forecasting, but their learned text signals are usually embedded inside a specific predictor as text embeddings, sentiment, or auxiliary features. They do not provide an event-type propagation graph that can be queried after training or reused as a monitoring artifact.

Large language models (LLMs) have also changed the financial text workflow. BloombergGPT, PIXIU, and FinBen show that finance-specific pretraining, instruction data, and benchmarks improve coverage of financial information extraction and textual analysis tasks [10]–[13]. These resources are useful for extracting structured signals from raw news. However, stronger language models do not by themselves specify when one type of financial event should become informative for another event or a future stock movement label. CausalLagStock therefore uses language models as grounding tools, while treating the lag-aware event-type propagation graph as the main object to be learned.

[1]Code is available at https://github.com/arisqutle/CausalLagStock

B. Event-Level Financial Text Understanding

Financial event extraction moves beyond document-level sentiment by identifying the concrete event category, participants, and attributes reported in text. Dedicated corpora and models established this setting: SENTiVENT provides a gold-standard corpus of company-specific economic events in financial news, annotated with event triggers, participant arguments, coreference, and attributes under an ACE/ERE-compatible scheme [14], and Doc2EDAG and knowledge-graph-enhanced methods study document-level financial event extraction [15], [16]. More recent work advances both the semantics and the extraction methodology. EFSA formulates event-level financial sentiment analysis with company–industry–event–sentiment quintuples [17]; the FINEED/MACK line studies text-to-event extraction from raw financial text and the difficulty of recognizing complex financial entities [18]; and large-language-model pipelines now target extraction at massive scale, using collaborative annotation and partitioning extraction over thousands of event types [19]. Together, these studies establish that financial text is best represented at the event level and that LLMs have become effective extraction engines.

Our work differs in how the extracted events are used. Existing tasks evaluate extraction or sentiment quality, and in stock prediction the resulting labels are typically consumed as static inputs to a predictor. CausalLagStock instead treats extraction as an observation layer rather than the final task: we ground news into a compact, market-calibrated 20-type schema and then learn a global map of directional event-type associations and preferred delays. We therefore adopt schema-constrained LLM extraction as a grounding tool because the learned object of interest is the temporal propagation structure over event types, not the extraction labels themselves.

C. Relational and Lagged Structure in Multi-Stock Prediction

Stock movement prediction is inherently relational because firms are connected through industries, supply chains, ownership, analyst coverage, and shared market conditions. Recent methods explicitly model this dependence. MASTER captures momentary and cross-time stock correlations with a market-guided Transformer [7]; StockMixer uses an efficient MLP architecture for indicator, time, and stock mixing [8]; and MDGNN constructs dynamic multi-relational graphs to represent evolving relations among stocks and associated entities [9]. For text-enhanced prediction, CMIN combines financial text with transfer-entropy-based causal attention over stock correlations [20]. CausalStock is closest to our setting because it performs lag-dependent causal discovery for news-driven multi-stock movement prediction and uses LLMs to denoise news [5].

These methods establish that directional and delayed relations matter for financial prediction. Their structural unit, however, is typically a stock, series, or stock relation. A learned edge can indicate that one asset or signal helps predict another, but it does not identify whether the relation is associated with earnings, litigation, macro data, analyst revision, or another event category. This limits interpretability and reuse when the stock universe changes. CausalStock and related methods therefore motivate our use of lagged structure, but our learned object is different: they learn causal or temporal dependencies mainly over stocks, stock relations, or stock-level signals, whereas CausalLagStock learns a shared graph over financial event types. This change of unit is central to the paper because it turns a stock-specific temporal relation into an event-level monitoring pattern.

D. Temporal Event Propagation and Lag-Aware Modeling

The idea that events can trigger later events has a long history in temporal point processes. Hawkes processes model mutually exciting events through time-decaying influence kernels [21]. Neural temporal point processes extend this idea with expressive sequence models, and recent work studies streaming adaptation and long-horizon event forecasting [22]–[25]. These models are well suited to irregular event streams and can capture delayed event dependencies.

The financial decision-support setting requires a different combination of properties. The event stream is heterogeneous and grounded in market context; the graph should be shared across firms and time periods; and the learned delays must be useful for chronological stock movement prediction. Existing point-process models usually focus on next-event likelihood or event sequence forecasting, while existing stock predictors usually operate at the asset level. CausalLagStock connects these directions by learning a sparse event-type propagation graph and injecting the learned delays into downstream forecasting and analyst monitoring. This design makes the propagation layer both predictive and inspectable.

III. Problem Setting and Method

Figure 1 presents an overview of CausalLagStock. The framework first constructs grounded event records, then learns a reusable lag-aware event-type propagation graph, and finally evaluates the graph in forward-in-time forecasting. The learned edges represent temporal predictive associations rather than interventional causality; they may reflect economic transmission, shared market exposure, reporting cycles, or chronological co-occurrence. The graph therefore serves as a compact and interpretable layer for forecasting and event monitoring.

A. Task Definition and Notation

Inputs. Let $\mathcal{U}$ be the stock universe. Stage 1 converts raw financial news into a chronological stream of structured event records, where each event i is a tuple

$$x_i = \big(e_i,\ c_i,\ \tau_i,\ f_i,\ \mathcal{V}_i\big), \tag{1}$$

with $e_i \in \mathbb{R}^{d_e}$ (d_e=768) a financial-text embedding [26], $c_i \in \{1, \dots, K\}$ an event type from a fixed schema of K=20 types,[2] τ_i the trading-day timestamp, $f_i \in \mathbb{R}^5$ an event-feature vector, and $\mathcal{V}_i \subseteq \mathcal{U}$ the affected-ticker set.

[2] Descriptions of the 20 event types are provided in the released implementation at `src/phase1/event_types.py`: https://github.com/arisqutle/CausalLagstock/blob/main/src/phase1/event_types.py.

The event-feature vector is defined as

$$f_i = \big[m_i,\ s_i,\ q_i^{\text{scope}},\ n_i,\ r_i^{\text{cred}}\big]^\top \in \mathbb{R}^5, \tag{2}$$

where m_i is the event-impact magnitude, s_i is surprise, q_i^{scope} is the numeric encoding of event scope, n_i is novelty, and r_i^{cred} is source credibility. The scope feature is numerically encoded as $q_i^{\text{scope}} = 0.25$ for single-stock events, 0.50 for sector-level events, 0.75 for market-wide events, and 1.00 for global events; missing or unrecognized scope values are encoded as 0.

For a target stock $u \in \mathcal{U}$ and a decision day t, the predictor may use *only* information available strictly before the decision: (i) the event window $X_{u,t} = (x_{i_1}, \dots, x_{i_L})$ formed by the L most recent events associated with u with $\tau_i \le t$, ordered by time; and (ii) the price window $P_{u,t} \in \mathbb{R}^{W \times 5}$ of the most recent W daily OHLCV (Open, High, Low, Close, and Volume) vectors ending at day t.

Prediction target. We predict the direction of the future close-to-close return of u. For a horizon of h trading days, let $p^{\text{c}}_{u,t}$ be the close price of u on day t and define the forward return $r^{(h)}_{u,t} = \big(p^{\text{c}}_{u,t+h} - p^{\text{c}}_{u,t}\big)/p^{\text{c}}_{u,t}$. With threshold $\theta = 0.5\%$, the categorical movement label is

$$y^{(h)}_{u,t} = \begin{cases} \text{up}, & r^{(h)}_{u,t} > \theta, \\ \text{down}, & r^{(h)}_{u,t} < -\theta, \\ \text{flat}, & \text{otherwise}, \end{cases} \tag{3}$$

encoded as $z^{(h)}_{u,t} \in \{0, 1, 2\}$ for the cross-entropy in Eq. (9).

The *primary* task (Stage 3) is next-trading-day three-class prediction (h=1): we learn a predictor

$$f_\phi :\ \big(X_{u,t},\ P_{u,t},\ u\big) \ \longmapsto\ \big(\hat{\pi}_{u,t} \in \Delta^2,\ \hat{m}_{u,t} \in \mathbb{R}\big), \tag{4}$$

where ϕ denotes the predictor parameters, $\hat{\pi}_{u,t}$ is a distribution over $\{\text{up}, \text{down}, \text{flat}\}$ on the probability simplex Δ^2, and $\hat{m}_{u,t}$ is an auxiliary signed-magnitude estimate.

Multi-horizon supervision (Stage 2). Because a propagation graph should not be learned from next-day supervision alone, Stage 2 supervises each sample b at several horizons $h \in \mathcal{H} = \{1, 3, 5, 10, 15\}$, giving a label vector $(z_b^{(h)})_{h \in \mathcal{H}}$; horizons whose future close is unavailable are masked out of the loss.

Learned object. The central object is a single *global* event-type propagation graph, shared across all stocks and time and summarized by two matrices,

$$A \in [0,1]^{K \times K}, \qquad T \in \mathbb{R}_{>0}^{K \times K}.$$

We adopt a source-to-target convention: for an earlier source event j and a later target event i (so $\tau_j < \tau_i$, with gap $\Delta\tau_{ij} = \tau_i - \tau_j$), $A[c_j, c_i]$ is the directed propagation strength and $T[c_j, c_i]$ the preferred trading-day delay of the relation $c_j \to c_i$. These matrices are estimated once in Stage 2 and reused unchanged by the Stage 3 predictors.

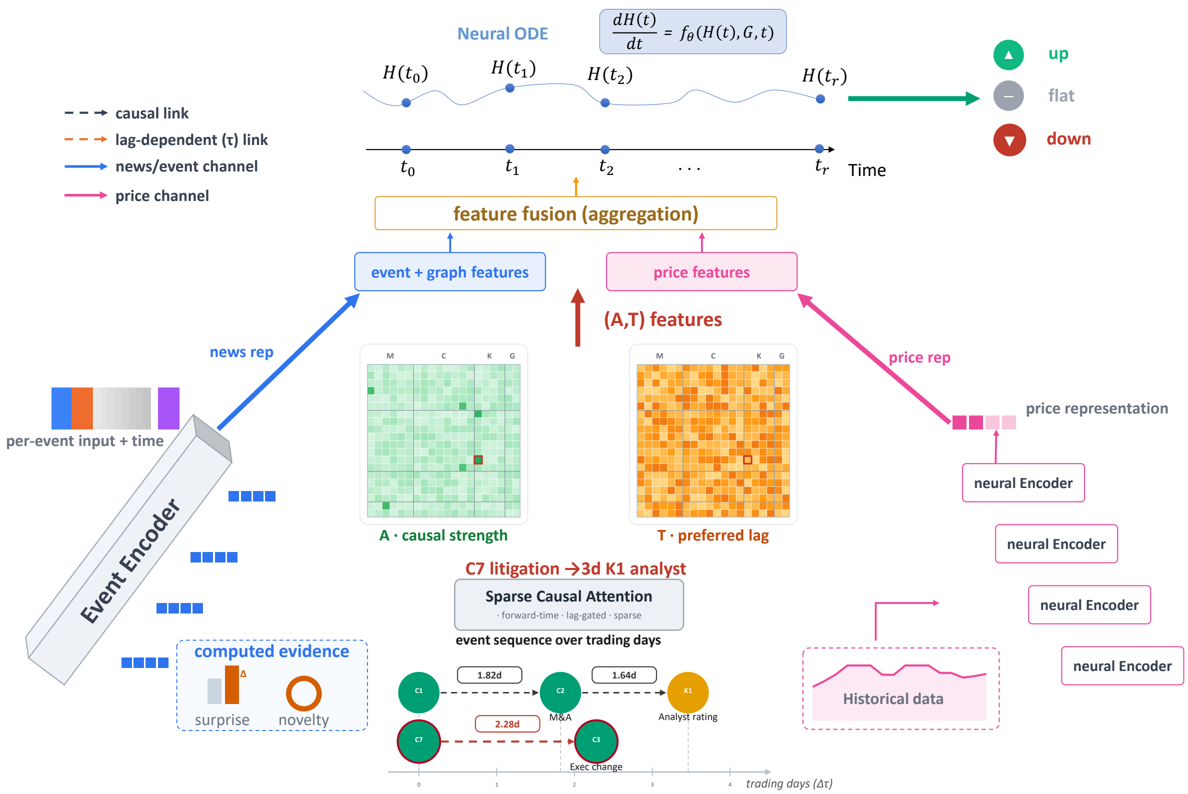

Fig. 1. High-level CausalLagStock pipeline. The left part grounds news into structured event records, the middle part learns a lag-aware event-type propagation graph, and the right part injects graph-derived signals into downstream prediction.

B. Stage 1: Grounded Event Construction

Stage 1 addresses a practical weakness of naive LLM-based annotation: a financial headline is often under-specified without market context. An earnings number is not meaningful without knowing whether it beat expectations, and an apparently important announcement may add little if it repeats recent same-ticker coverage. Stage 1 maps each raw news item to the grounded event record in Eq. (1) in two complementary steps. First, a large language model produces a richly annotated *silver* label, with two deterministic market/text signals supplied as contextual evidence; second, a pretrained FinBERT encoder produces a contextual representation of the headline. The structured LLM annotations and FinBERT embeddings are then used directly by the downstream causal-graph and stock-prediction stages.

Event schema and annotation targets. The schema has $K{=}20$ event types in four families: macroeconomic (M1–M6), company (C1–C8), market-knowledge (K1–K4), and geopolitical (G1–G2), plus a NONE class for items with no financial event. Each annotation contains the event type c_i, a subject/action/object triple, a magnitude score $m_i \in [-10, 10]$, a polarity in {positive, negative, neutral, mixed}, a scope in {single-stock, sector, market, global}, a source-credibility score in $[0, 1]$, the surprise s_i and novelty n_i defined next, and the affected-ticker set $\mathcal{V}_i$. The subject/action/object triple is retained for interpretation and quality auditing, whereas the downstream models primarily use the event type, magnitude, polarity, affected tickers, and numeric impact-profile fields.

Deterministic surprise. A reported figure is informative only relative to expectation, so we compute an auxiliary surprise signal *before* prompting. When the text or aligned market context exposes a reported value a_i and a consensus/forecast value $\bar{a}_i$, surprise is the clamped expectation gap

$$s_i = \mathrm{clip}\left(\frac{a_i - \bar{a}_i}{\max(|\bar{a}_i|, \varepsilon)},\ -1,\ 1\right), \tag{5}$$

where $\varepsilon > 0$ prevents division by zero. When no expectation is exposed, s_i falls back to the prior 3-trading-day return ending strictly before the event when such market context is available; otherwise, it is left undefined rather than fabricated.

Deterministic novelty. To discount repeated coverage, we embed the current item and its 10 most recent same-ticker predecessors $\mathcal{P}_i$ with FinBERT and count near-duplicates:

$$n_i = \frac{1}{1 + |\{\, j \in \mathcal{P}_i : \cos(z_i, z_j) \geq \kappa \,\}|} \in (0, 1], \quad \kappa = 0.85, \tag{6}$$

where z_i denotes the FinBERT representation used for novelty comparison. A stopword-filtered lexical Jaccard score replaces cosine similarity when embeddings are unavailable, and $n_i{=}1$ when $\mathcal{P}_i = \varnothing$.

Schema-constrained LLM annotation. We prompt a strong instruction-following LLM through an API under a strict schema, decoding at temperature 0.1 and emitting a single JSON record per item. The input contains the headline and, when article input is enabled, an article excerpt of up to 1,200 characters. The precomputed (s_i, n_i) are injected as soft contextual evidence: polarity and scope are judged from the text, surprise and novelty are informed by the corresponding auxiliary signals when available, and credibility is judged from source provenance (e.g. tier-1 wire and regulatory sources receive the highest scores). For items reporting several events, the most salient one is retained. The resulting JSON records are stored and directly reused by the downstream stages, avoiding repeated API calls for subsequent graph-learning and prediction experiments.

FinBERT text representation. Independently of the structured LLM annotation, each headline is encoded using a pretrained FinBERT model with hidden size 768 and maximum sequence length 128. The final- layer [CLS] state is retained as the 768-d textual embedding e_i. In the main pipeline, FinBERT is used as a pretrained feature encoder rather than as a separately distilled event extractor. The LLM annotation supplies the structured event fields, while FinBERT supplies the textual representation. Specifically, the LLM supplies c_i, f_i, and $\mathcal{V}_i$, whereas FinBERT supplies e_i. Together, these components instantiate the grounded event record x_i defined in Eq. (1).

Silver-label quality control. To check the reliability of the grounded event labels, we manually audit a stratified sample across event families and polarity classes. The audit focuses on event type, affected ticker, polarity, and whether the assigned magnitude is directionally plausible. This audit is used only as a quality check for the observation layer; the downstream graph is learned from chronological prediction supervision.

C. Stage 2: Sparse Lag-Aware Event-Type Discovery

Stage 2 learns a reusable event-type propagation graph from chronological financial event sequences. Its purpose is to identify which event categories tend to precede later event-driven signals or price reactions, and which trading-day delay is most useful for prediction. The learned graph is defined over event types rather than individual news items or stocks, so it can be shared across samples and inspected after training.

Let the input event sequence be

$$\mathcal{E} = \{(e_i, c_i, \tau_i)\}_{i=1}^{L},$$

where $e_i \in \mathbb{R}^{768}$ is the text embedding, $c_i \in \{1, \dots, K\}$ is the event type with $K = 20$, and τ_i is the trading-day timestamp. Stage 2 learns two global matrices:

$$A \in [0,1]^{K\times K}, \qquad T \in \mathbb{R}_{+}^{K\times K}.$$

Here $A[a, b]$ denotes the propagation strength from source event type a to target event type b, while $T[a, b]$ denotes the preferred delay in trading days. These matrices are shared across the full dataset and are intended to serve as a compact event-type knowledge layer.

Each event is encoded by its text, event-type, and time representation into a hidden state $h_i \in \mathbb{R}^d$. Inspired by time-aware self-attention for temporal event sequences [27], we introduce a lag-aware attention mechanism with event-type-specific propagation strength and lag bias. For a later target event i and an earlier source event j, the attention score is

$$\begin{aligned} s_{ij} = &\frac{(W_Q h_i)^\top (W_K h_j)}{\sqrt{d}} + \log(A[c_j, c_i] + \epsilon) \\ &- \frac{(|\tau_i - \tau_j| - T[c_j, c_i])^2}{2\sigma^2} + M_{ij}, \end{aligned} \tag{7}$$

where W_Q and W_K are learned query/key projections, d is the hidden dimension used for attention scaling, M_{ij} is a temporal mask that blocks future-to-past information flow, σ controls the width of the lag kernel, and $\epsilon > 0$ is a small constant for numerical stability. The content-similarity term captures instance-level compatibility, the strength term promotes frequently useful event-type relations, and the lag term favors pairs whose observed temporal distance matches the learned preferred delay.

This design gives Stage 2 three properties. First, propagation is directional because only earlier events can contribute to later events. Second, propagation is delay-aware because each event-type pair has its own preferred lag. Third, the learned graph is global and sparse, making it easier to inspect than dense sample-specific attention.

D. Multi-Horizon Supervision and Graph Regularization

Stage 2 is supervised by future stock movement labels at multiple trading-day horizons. For each event sequence, we compute forward close-to-close returns for the candidate horizon set

$$\mathcal{H} = \{1, 3, 5, 10, 15\}.$$

Using the same return definition as Eq. (3), each return is converted into a three-class direction label:

$$z^{(h)} = \begin{cases} \text{UP}, & r^{(h)} > 0.5\%, \\ \text{DOWN}, & r^{(h)} < -0.5\%, \\ \text{FLAT}, & \text{otherwise}. \end{cases}$$

The Stage 2 predictor emits three-class logits for every horizon $h \in \mathcal{H}$, and $p_{b,h}$ below denotes the corresponding softmax distribution for sample b and horizon h.

To connect graph discovery with horizon supervision, Stage 2 derives a horizon weight from the learned graph. For each training sample, we define q as the event type of the most recent event in the event window, which serves as the local target type for horizon weighting. The unnormalized weight for horizon h is

$$w_h \propto \sum_{j=1}^{L} A[c_j, q] \exp\left(-\frac{(T[c_j, q] - h)^2}{2\sigma_h^2}\right), \tag{8}$$

where σ_h controls horizon-weight smoothing. The weights are normalized over valid horizons to obtain $w_{b,h}$. Thus, short-lag event relations emphasize short-horizon labels, while longer-lag relations shift supervision toward longer horizons.

Let $M_{b,h} = \mathbf{1}[z_b^{(h)} \neq -100]$ denote whether the horizon label is valid. The predictive loss is implemented as

$$\mathcal{L}_{\text{pred}} = \frac{\sum_b \sum_{h\in\mathcal{H}} M_{b,h}\, w_{b,h}\, \alpha_{b,h}\, \ell_{b,h}}{\sum_b \sum_{h\in\mathcal{H}} M_{b,h}\, w_{b,h}\, \alpha_{b,h}}, \tag{9}$$

where $\ell_{b,h} = -\log p_{b,h}(z_b^{(h)})$ is the negative log-likelihood assigned to the true direction label of sample b at horizon h. Here, $z_b^{(h)}$ is the corresponding three-class direction label, $w_{b,h}$ is the normalized horizon relevance weight derived from the learned lag-aware propagation graph, and $\alpha_{b,h}$ is a class-frequency reweighting term. The mask $M_{b,h}$ indicates whether the horizon label is valid and removes horizons whose labels cannot be computed because the required future close price is unavailable. The denominator normalizes by the total effective weight, so batches with different numbers of valid horizons remain comparable.

The graph regularizer is

$$\begin{aligned}\mathcal{R}_{\text{graph}} =& \lambda_{\text{sparse}}\|A\|_1 + \lambda_{\text{dag}} h(A)^2 \\ &- \lambda_{\text{var}} \text{Var}(A).\end{aligned} \tag{10}$$

The total Stage 2 objective is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \rho\, \mathcal{R}_{\text{graph}}. \tag{11}$$

where ρ is the overall graph-regularization weight. The graph regularizer $\mathcal{R}_{\text{graph}}$ combines three terms: $\|A\|_1$ encourages weak event-type edges to shrink, $h(A)$ is a soft NOTEARS-style cycle-discouragement penalty [28], and the negative variance term prevents the learned graph from collapsing into an uninformative nearly constant matrix. The coefficients λ_{sparse}, λ_{dag}, and λ_{var} control the three regularization terms. We use $h(A) = \text{tr}(\exp(A \circ A)) - K$, where $\circ$ denotes elementwise multiplication and $\exp(\cdot)$ denotes the matrix exponential. The cycle-discouragement term is used as a soft readability bias rather than a hard directed-acyclic-graph constraint. The learned graph may still retain reciprocal relations when they are strongly supported by chronological evidence. The computational complexity of Stage 2 is dominated by the stacked causal-lag attention layers. Given N samples, sequence length L, hidden dimension D, and R attention layers, the overall training complexity is approximately

$$\mathcal{O}(ENR(LD^2 + L^2 D))$$

where E denotes the number of training epochs.

Because the data are observational and financial narratives may contain reciprocal associations, the graph is interpreted as a directional event-propagation structure for prediction rather than as definitive interventional causality. The selected Stage 2 output is the graph checkpoint with the best validation horizon accuracy. It contains the learned propagation strength matrix A and lag matrix T, which are passed to Stage 3.

E. Stage 3: Graph-Enhanced Downstream Prediction

Once the event-type graph is learned, Stage 3 tests its operational value in downstream prediction using the frozen propagation strength matrix A and lag matrix T.

Transparent tabular predictor. A histogram gradient boosting (HGB) classifier uses the graph as static graph-derived features. For each sample, we construct event-window summaries including event-type counts, signed and absolute magnitude summaries, the last event type, event-profile summaries, target-stock identity, and price-window statistics. The graph-derived part is formed by projecting raw event-type count vector $\text{counts} \in \mathbb{R}^K$ and magnitude-weighted count vector $\text{weighted} \in \mathbb{R}^K$ through the learned matrices:

$$[\,\text{counts}A,\ \text{counts}T,\ \text{weighted}A,\ \text{weighted}T\,].$$

The projections through T summarize the delay profile induced by the event window. These features are concatenated with price, event, and stock features. This tabular variant makes the graph contribution auditable and supports no-graph, partial-graph, randomized-graph, and reversed-graph comparisons.

Graph-enhanced Neural ODE. Inspired by Neural Ordinary Differential Equations (Neural ODEs) [29], the second predictor treats the learned graph as a continuous-time coupling prior rather than as a fixed feature summary. Each event is first mapped to an impact vector using its text embedding, event-type embedding, event-impact magnitude, and four-dimensional event profile. Event impact vectors are then aggregated into type-specific latent states using event-type indicators and recency weights.

Let $S \in \mathbb{R}^{K\times d_s}$ denote the matrix of type-specific latent states, let t denote continuous propagation time, let $f(\cdot)$ be a learned dynamics network, and let δ be a learned decay parameter whose softplus transformation is nonnegative. The event-type states evolve according to

$$\frac{dS}{dt} = f([\,S,\ \text{agg}(S,t),\ t\,]) - \text{softplus}(\delta)\, S, \tag{12}$$

where the graph-conditioned aggregation is

$$\text{agg}(S,t)[q] = \sum_p A[p,q]\, \text{sigmoid}\big(\beta(t - T[p,q])\big)\, S[p]. \tag{13}$$

Here $S[p]$ is the latent state of source event type p, $\text{agg}(S,t)[q]$ is the aggregated incoming signal for target event type q, and β controls how sharply the delayed signal activates around the preferred delay $T[p,q]$. Thus, source event type p contributes to target event type q according to both its learned propagation strength and its learned preferred delay.

After ODE propagation, the final event-type states are mapped to stock states through a learned event–stock affinity matrix. A lightweight stock-graph attention layer refines these stock states. In parallel, a GRU encodes the OHLCV price window. The final representation concatenates the target-stock state, an attention-pooled stock representation, and the price representation. A prediction head outputs the three-class movement label and an auxiliary signed-return magnitude estimate.

IV. Experiments

A. Datasets and Inputs

The experiments combine structured financial news events, financial text embeddings, and aligned price histories. We evaluate the model on FNSPID and an aligned CMIN-US setting. FNSPID is a large-scale time-aligned financial news and stock price dataset for stock movement prediction, containing 29.7 million price records and 15.7 million news records for 4,775 S&P 500-related companies from 1999 to 2023 [6]. For tractability, we use a labeled FNSPID subset with 50,740 event-labeled records from 2021-01-01 to 2023-12-16, which yields 15,295 aligned prediction examples across 184 tickers after event-price matching.

CMIN is a stock-movement benchmark released with the Causality-Guided Multi-Memory Interaction Network [20]. We use its CMIN-US branch, which covers U.S. financial texts and historical prices from 2018-01-01 to 2021-12-31. After alignment with available event labels, embeddings, and price series, our CMIN-US experiment contains 107 stocks.

Each sample uses a recent event sequence, a target stock identifier, and an OHLCV price window. The current Stage 3 datasets include exact label-end timestamps for chronological purging.

The processed Stage 3 datasets show a clear class imbalance, mainly because the Flat class is much smaller than the Up and Down classes, as shown in Table I. Therefore, evaluation should not rely only on accuracy, and measures such as Macro-F1, Weighted-F1, and MCC are useful for assessing performance under this imbalance.

TABLE I
CLASS DISTRIBUTION OF THE PROCESSED STAGE 3 DATASETS BEFORE CHRONOLOGICAL SPLITTING AND LABEL-END PURGING.

Dataset	Up	Down	Flat	Total
FNSPID	7,154	6,739	1,402	15,295
CMIN	21,624	17,566	5,771	44,961

B. Experimental Protocol

Our experiments follow a deployment-oriented chronological protocol: the predictor is trained on earlier periods, selected on later validation periods, and evaluated on held-out future periods only. The updated implementation groups identical timestamps and applies a label-end purge so that training examples whose prediction horizon reaches into the validation period are removed. Stage 2 graph learning uses event sequences of length 32 together with labels at horizons $\{1, 3, 5, 10, 15\}$ trading days. Stage 3 uses the same chronological principle after event-price alignment and uses exact label-end timestamps when available.

For the completed FNSPID Stage 3 run, the chronological split contains 10,452 training samples, 1,940 validation samples, and 2,273 test samples. For the completed CMIN Stage 3 run, the chronological split contains 31,268 training samples, 6,587 validation samples, and 6,745 test samples. Both runs use exact label-end purging. All Stage 2 and Stage 3 work was conducted on a workstation with NVIDIA RTX 5090 GPU and 25 vCPU Intel(R) Xeon(R) Platinum 8470Q. Stage 2 training took approximately 14 minutes and 32 seconds, while training the Stage 3 Neural ODE model took approximately 13–19 minutes per random seed.

For reproducibility, we summarize the main optimization settings here. Stage 2 uses a train ratio of 0.70, batch size 16, AdamW optimization, learning rate 1×10^{-5}, weight decay 0.01, and early stopping on validation horizon accuracy with patience 4. Stage 3 Neural ODE training uses five random seeds, batch size 128, AdamW with learning rate 3×10^{-4}, weight decay 1×10^{-4}, 25 epochs, focal loss with $\gamma = 1.5$, balanced class weighting with power 0.6, and MCC for model selection.

C. Main Results

The evaluation measures include accuracy, macro precision, macro recall, macro-F1, weighted-F1, and Matthews correlation coefficient (MCC). Weighted-F1 averages class-wise F1 scores weighted by each class's sample size, measuring overall classification performance under class imbalance. Table II reports the completed three-class Stage 3 results. CausalLagStock reports five random seeds, while the reproduced baselines report three runs due to computational cost; the comparison is therefore a chronological reference comparison under the available aligned settings rather than a perfectly harmonized leaderboard.

Table II shows that CausalLagStock improves accuracy and weighted-F1 on both datasets under chronological evaluation. These gains indicate that the learned lag-aware propagation graph helps the predictor make more reliable predictions on the dominant test distribution. However, the improvements are not uniform across all evaluation measures. Macro-level measures and MCC remain modest, especially on CMIN, because the Flat class is small and threshold-sensitive. We do not interpret the modest MCC as a failure of the propagation graph alone. Instead, it reflects the combined difficulty of minority-class Flat recognition, threshold-sensitive labeling, and temporal distribution shift under chronological evaluation. We therefore interpret the results as evidence that event-type propagation improves overall predictive reliability, while balanced three-class discrimination remains an open challenge.

D. Ablation, Placebo, and Error Analysis

Table III reports the ablation experiment conducted on the CMIN dataset. The complete lag-aware propagation graph achieves the best accuracy among the Neural ODE variants, with an accuracy of 0.4246 and a Macro-F1 of 0.3172. Compared with the no-graph variant, it improves accuracy and Macro-F1 by 1.53 and 0.34 percentage points, respectively. Removing either component degrades performance. In particular, the *T only* variant achieves the lowest accuracy, suggesting that temporal-lag information alone is insufficient without the corresponding propagation strengths. Although the *A only* variant obtains a similar accuracy of 0.4233, its lower Macro-F1 (0.3009) indicates less balanced predictions across the three movement classes.

The random-graph trial performs similarly to the no-graph variant and remains below the full model in both measures.

TABLE II

CHRONOLOGICAL THREE-CLASS STOCK MOVEMENT PREDICTION UNDER LABEL-END PURGING. "SEEDS" IS THE NUMBER OF RANDOM SEEDS USED FOR REPORTING.

Model	Dataset	Seeds	Acc.	Macro-P	Macro-R	Macro-F1	Weighted-F1	MCC
StockNet [1]	CMIN	3	0.3667 ± 0.0186	0.1593 ± 0.0580	0.3332 ± 0.0002	0.1795 ± 0.0055	0.1978 ± 0.0157	-0.0014 ± 0.0025
Causality-driven Transformer [4]	CMIN	3	0.3448 ± 0.0037	0.3445 ± 0.0038	0.3452 ± 0.0036	0.3436 ± 0.0036	0.3435 ± 0.0037	0.0175 ± 0.0025
CausalLagStock w/ Neural ODE (Ours)	CMIN	5	$\mathbf{0.4246 \pm 0.0298}$	0.3415 ± 0.0125	0.3373 ± 0.0110	0.3172 ± 0.0131	$\mathbf{0.4009 \pm 0.0122}$	0.0075 ± 0.0243
StockNet	FNSPID	3	0.3689 ± 0.0177	0.2045 ± 0.0738	0.3312 ± 0.0025	0.2236 ± 0.0522	0.2509 ± 0.0682	-0.0072 ± 0.0092
Causality-driven Transformer	FNSPID	3	0.3577 ± 0.0009	0.3579 ± 0.0005	0.3623 ± 0.0005	0.3559 ± 0.0008	0.3579 ± 0.0013	0.0368 ± 0.0008
CausalLagStock w/ Neural ODE (Ours)	FNSPID	5	$\mathbf{0.4939 \pm 0.0131}$	$\mathbf{0.3715 \pm 0.0129}$	0.3601 ± 0.0146	0.3513 ± 0.0236	$\mathbf{0.4698 \pm 0.0192}$	$\mathbf{0.0511 \pm 0.0313}$

TABLE III

LAG-AWARE PROPAGATION GRAPH ABLATION AND PLACEBO COMPARISON. THE PLACEBO TESTS WHETHER THE GAINS COME FROM THE LEARNED GRAPH RATHER THAN ARBITRARY GRAPH-DERIVED FEATURES.

Variant	Model	Seeds	Dim.	Acc.	Macro-F1
No graph	Neural ODE	5	seq.	0.4093 ± 0.0485	0.3138 ± 0.0143
A only	Neural ODE	5	seq.	0.4233 ± 0.0363	0.3009 ± 0.0349
T only	Neural ODE	5	seq.	0.3941 ± 0.0410	0.3119 ± 0.0186
Random graph	Neural ODE	5	seq.	0.4096 ± 0.0467	0.3129 ± 0.0150
Full lag-aware graph	Neural ODE	5	seq.	$\mathbf{0.4246 \pm 0.0298}$	0.3172 ± 0.0131
Full lag-aware graph	HGB	5	280	0.4089 ± 0.0091	$\mathbf{0.3319 \pm 0.0088}$

TABLE IV

AVERAGE TEST CONFUSION MATRIX FOR THE GRAPH-ENHANCED NEURAL ODE MODEL. ROWS CORRESPOND TO TRUE CLASSES AND COLUMNS CORRESPOND TO PREDICTED CLASSES.

True / Pred.	Up	Down	Flat
Up	1686.4	1416.2	220.4
Down	1286.0	1119.0	165.0
Flat	435.6	357.6	58.8

This indicates that the improvement does not merely result from adding graph-structured inputs or increasing model complexity. Instead, the learned lagged predictive associations and their associated temporal lags need to be jointly modeled.

The HGB model presents a useful applied trade-off on CMIN. It achieves lower accuracy than the full Neural ODE model (0.4089 versus 0.4246), but obtains a higher Macro-F1 (0.3319 versus 0.3172). Thus, HGB is more transparent and provides more balanced predictions across the three classes, whereas the Neural ODE model correctly predicts a larger proportion of the test samples. Overall, the CMIN ablation results support the joint contribution of learned propagation strengths and temporal lags, while highlighting the remaining challenge of stable minority-class prediction.

The confusion matrix shows that most Flat examples are absorbed into Up or Down, confirming that the main error source is threshold-sensitive minority-class recognition rather than a complete failure of directional prediction.

V. INTERPRETING THE LEARNED LAG-AWARE EVENT PROPAGATION GRAPH

The learned event-type propagation graph is the main interpretable mining output of CausalLagStock. Although the downstream predictor outputs stock movement labels, the mining objective is broader: the framework extracts recurring temporal regularities from timestamped financial news and organizes them into a reusable event-type graph. Unlike instance-level attention maps, this graph is shared across samples and represents directed relations between event types. Each edge is associated with both a learned propagation strength and a preferred trading-day lag, making the graph inspectable as a compact knowledge layer for event monitoring, horizon-aware risk analysis, and stress testing. The representative edges in this section are not standalone evidence that the graph is correct. The full $K \times K$ graph is evaluated at scale through the chronological prediction, ablation, and random-graph placebo experiments in Section IV; the edge table and visualization provide semantic inspection of high-strength relations, while exhaustive expert validation is left for future domain review.

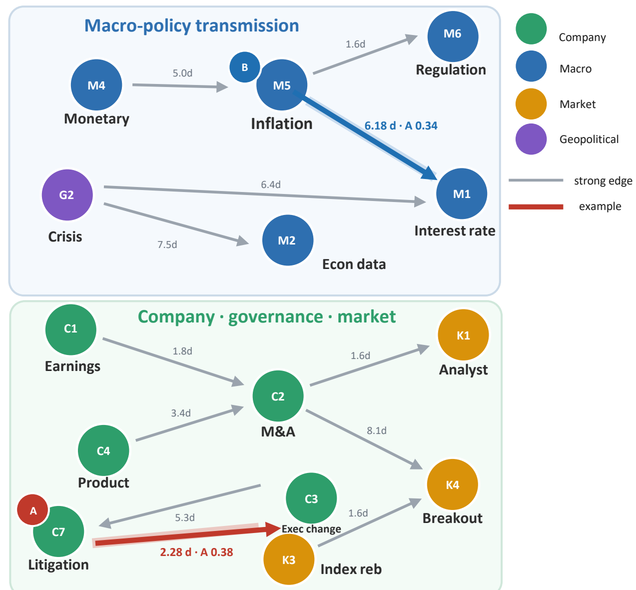

Fig. 2. Edge-level demo derived from Table V. Nodes denote event types, gray arrows denote high-strength learned relations, and highlighted arrows show how detected events can trigger horizon-specific monitoring targets, with learned strength A and preferred delay T. The figure is an analyst-facing summary of high-strength delayed associations mined from chronological news and price supervision, rather than a proof of interventional causality.

A. Event-Type Relations as Lagged Mined Patterns

The basic unit mined by the graph is an event-type relation, rather than a stock-to-stock relation or a sentence-level sentiment relation. A directed edge $p \rightarrow q$ indicates that occurrences of event type p provide useful lagged information for later representations associated with event type q and

TABLE V

REPRESENTATIVE LEARNED EVENT-TYPE PROPAGATION RELATIONS. "STRENGTH" IS THE LEARNED DIRECTED EDGE WEIGHT AND "LAG" IS THE PREFERRED DELAY IN TRADING DAYS.

Edge	Financial meaning	Strength	Lag	Interpretation
M5→M1	Inflation data → interest rates	0.3447	6.18d	Inflation news shifts expectations about rate paths over multiple days.
M4→M5	Monetary policy → inflation data	0.3404	5.04d	Policy communication and inflation narratives co-propagate over the news cycle.
M5→M6	Inflation data → regulation	0.3649	1.57d	Price pressure can be followed by regulatory or policy reaction.
C1→C2	Earnings → M&A	0.3659	1.82d	Strong or weak earnings can reshape strategic transaction expectations.
C4→C2	Product launch → M&A	0.3721	3.38d	Product news can alter consolidation expectations within a sector.
C2→K1	M&A → analyst ratings	0.3490	1.64d	Deal announcements often trigger analyst reassessment.
C7→C3	Litigation/fines → executive changes	0.3786	2.28d	Legal or regulatory shocks often precede governance instability.
C3→C7	Executive changes → litigation/fines	0.3668	5.31d	Departures may be followed by delayed compliance disclosure.
K3→K4	Index rebalance → technical breakout	0.3452	1.60d	Passive-flow changes may affect short-term price patterns.
C2→K4	M&A → technical breakout	0.3227	8.07d	Deal news can gradually propagate into price-trend signals.
G2→M1	Crisis → interest rates	0.3856	6.44d	External shocks can alter expectations for rates and policy response.
G2→M2	Crisis → economic data	0.3586	7.49d	Shocks increase attention to later macroeconomic releases.

the downstream movement label. The edge weight $A[p,q]$ measures how strongly the learned graph relies on this relation, while $T[p,q]$ records the trading-day delay at which the relation is most informative. Therefore, the graph should be interpreted as a lag-aware summary of temporally ordered predictive patterns: it describes which event categories tend to precede later event or price-relevant signals, and when such propagation tends to become useful.

However, it does not imply that intervening on event type p would necessarily change event type q or the stock movement outcome. Because the inputs are observational news and price records, an edge may reflect direct economic transmission, common exposure to a shared market regime, reporting cycles, analyst coverage patterns, or repeated co-occurrence under temporal ordering. The practical value of the graph is that it turns diffuse news streams into a compact set of auditable propagation patterns.

B. Propagation Pathways in the Learned Graph

Figure 2 illustrates how the learned graph converts an extracted event into a delayed propagation pathway. Each node denotes an event type, and node color indicates its broader event family, such as company, macro, market, or geopolitical events. Gray arrows denote high-strength relations selected from the global graph, while highlighted arrows show retrieved paths with their preferred lags and learned strengths.

The first highlighted path illustrates macro-policy transmission. After Stage 1 maps a headline such as an unexpectedly high consumer-price report to an inflation event (M5), the learned graph retrieves the relation M5→M1. In this context, propagation means that the inflation event type has repeatedly provided useful lagged information for later interest-rate-related signals, with a preferred delay of about 6.18 trading days. The second highlighted path illustrates a company-governance pattern. When a headline is mapped to litigation or fines (C7), the graph retrieves a relation toward executive changes (C3). The preferred delay of about 2.28 trading days indicates a short monitoring window in which legal-risk news has often preceded governance-related signals in the chronological data.

C. Representative Propagation Families

Table V reports representative high-strength learned propagation relations from the learned event-type graph. For each relation, we show the directed event pair, its financial meaning, learned strength, preferred trading-day lag, and a short domain interpretation. The table exposes representative learned relations so that the mined structure can be inspected for financial plausibility. These relations should be read as mined delayed predictive patterns, not as causal rules.

Several coherent propagation families appear in the mined edge set. Macro-policy relations such as M5→M1 and G2→M1 suggest that inflation or crisis news is followed by interest-rate-related signals over a multi-day window. This is financially plausible because macro news often propagates through policy commentary, rate-path revisions, and follow-up macro coverage rather than affecting market interpretation only at the release time. Company-strategy relations such as C1→C2, C4→C2, and C2→K1 indicate that earnings, product, and merger-related events may trigger later strategic expectations or analyst reassessments. Governance-risk relations such as C7→C3 and C3→C7 capture temporal links between legal shocks and executive changes. Market-technical relations such as K3→K4 and C2→K4 connect event-driven flows or deal news to later breakout-style price patterns.

VI. CONCLUSION

We presented CausalLagStock, a framework that mines lag-aware event-type propagation graphs from financial news and market observations. The learned graph serves as an intermediate layer between raw news and predictive financial analysis, encoding directed propagation strengths and preferred trading-day delays between event types. Experiments on FNSPID and CMIN showed that the framework improves accuracy and weighted-F1 under chronological evaluation and produces interpretable delayed event pathways. At the same time, macro-level measures and MCC show that balanced recognition of threshold-sensitive minority classes remains challenging. Future work will improve minority-class calibration, validate the learned propagation graph with expert financial judgment, and extend the framework to broader markets and event schemas.

REFERENCES

[1] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '18, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 1970–1979.

[2] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, ser. WSDM '18. Marina Del Rey, CA, USA: Association for Computing Machinery, 2018, pp. 261–269.

[3] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, ser. IJCAI '15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 2327–2333.

[4] A. Díaz Berenguer, Y. Da, M. Bossa, M. Oveneke, and H. Sahli, "Causality-driven multivariate stock movement forecasting," *PLOS ONE*, vol. 19, no. 4, p. e0302197, Apr. 2024.

[5] S. Li, Y. Sun, Y. Lin, X. Gao, S. Shang, and R. Yan, "CausalStock: Deep end-to-end causal discovery for news-driven multi-stock movement prediction," in *Advances in Neural Information Processing Systems*, ser. NeurIPS '24, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Vancouver, BC, Canada: Curran Associates, Inc., 2024, pp. 47 432–47 454.

[6] Z. Dong, X. Fan, and Z. Peng, "FNSPID: A comprehensive financial news dataset in time series," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '24. Barcelona, Spain: Association for Computing Machinery, 2024, pp. 4918–4927.

[7] T. Li, Z. Liu, Y. Shen, X. Wang, H. Chen, and S. Huang, "MASTER: Market-guided stock transformer for stock price forecasting," in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI '24. Vancouver, BC, Canada: AAAI Press, 2024.

[8] J. Fan and Y. Shen, "StockMixer: A simple yet strong MLP-based architecture for stock price forecasting," in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI '24. Vancouver, BC, Canada: AAAI Press, 2024.

[9] H. Qian, H. Zhou, Q. Zhao, H. Chen, H. Yao, J. Wang, Z. Liu, F. Yu, Z. Zhang, and J. Zhou, "MDGNN: Multi-relational dynamic graph neural network for comprehensive and dynamic stock investment prediction," in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI '24. Vancouver, BC, Canada: AAAI Press, 2024.

[10] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. S. Rosenberg, and G. Mann, "BloombergGPT: A large language model for finance," *ArXiv preprint*, vol. arXiv:2303.17564, 2023.

[11] Q. Xie, W. Han, X. Zhang, Y. Lai, M. Peng, A. Lopez-Lira, and J. Huang, "PIXIU: A comprehensive benchmark, instruction dataset and large language model for finance," in *Advances in Neural Information Processing Systems*, ser. NeurIPS '23, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. New Orleans, LA, USA: Curran Associates, Inc., 2023, pp. 33 469–33 484.

[12] Q. Xie, W. Han, Z. Chen, R. Xiang, X. Zhang, Y. He, M. Xiao, D. Li, Y. Dai, D. Feng, Y. Xu, H. Kang, Z. Kuang, C. Yuan, K. Yang, Z. Luo, T. Zhang, Z. Liu, G. Xiong, Z. Deng, Y. Jiang, Z. Yao, H. Li, Y. Yu, G. Hu, J. Huang, X.-Y. Liu, A. Lopez-Lira, B. Wang, Y. Lai, H. Wang, M. Peng, S. Ananiadou, and J. Huang, "FinBen: A holistic financial benchmark for large language models," in *Advances in Neural Information Processing Systems*, ser. NeurIPS '24, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Vancouver, BC, Canada: Curran Associates, Inc., 2024, pp. 95 716–95 743.

[13] Y. Zhou, F. Zhang, Y. Chen, H. Zhang, P. Nakov, and Z. Xie, "FinCARDS: Card-based analyst reranking for financial document question answering," in *Findings of the Association for Computational Linguistics: ACL 2026*, ser. ACL '26. San Diego, California, United States: Association for Computational Linguistics, 2026, pp. 24 836–24 852.

[14] G. Jacobs and V. Hoste, "SENTiVENT: Enabling supervised information extraction of company-specific events in economic and financial news," *Language Resources and Evaluation*, vol. 56, no. 1, pp. 225–257, Mar. 2022.

[15] S. Zheng, W. Cao, W. Xu, and J. Bian, "Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, ser. EMNLP-IJCNLP '19, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 337–346.

[16] K. Guo, T. Jiang, and H. Zhang, "Knowledge graph enhanced event extraction in financial documents," in *Proceedings of the 2020 IEEE International Conference on Big Data*, ser. IEEE BigData '20. Atlanta, GA, USA: IEEE, 2020, pp. 1322–1329.

[17] T. Chen, Y. Zhang, G. Yu, D. Zhang, L. Zeng, Q. He, and X. Ao, "EFSA: Towards event-level financial sentiment analysis," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ser. ACL '24, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 7455–7467.

[18] Y. Huang, N. Hu, K. Li, N. Wang, and Z. Lin, "Extracting financial events from raw texts via matrix chunking," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, ser. LREC-COLING '24, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds. Torino, Italia: ELRA and ICCL, 2024, pp. 7035–7044.

[19] W. Liu, Z. Li, L. Bai, Y. Zuo, D. Xu, X. Jin, J. Guo, and X. Cheng, "Towards event extraction with massive types: LLM-based collaborative annotation and partitioning extraction," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '25, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Suzhou, China: Association for Computational Linguistics, 2025, pp. 34 377–34 399.

[20] D. Luo, W. Liao, S. Li, X. Cheng, and R. Yan, "Causality-guided multi-memory interaction network for multivariate stock price movement prediction," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ser. ACL '23, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 12 164–12 176.

[21] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, Apr. 1971.

[22] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. San Francisco, CA, USA: Association for Computing Machinery, 2016, pp. 1555–1564.

[23] H. Mei and J. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," in *Advances in Neural Information Processing Systems*, ser. NeurIPS '17, vol. 30. Long Beach, CA, USA: Curran Associates, Inc., 2017, pp. 6754–6764.

[24] S. Xue, Y. Wang, Z. Chu, X. Shi, C. Jiang, H. Hao, G. Jiang, X. Feng, J. Y. Zhang, and J. Zhou, "Prompt-augmented temporal point process for streaming event sequence," in *Advances in Neural Information Processing Systems*, ser. NeurIPS '23, vol. 36. New Orleans, LA, USA: Curran Associates, Inc., 2023.

[25] M. Zeng, F. Regol, and M. Coates, "Interacting diffusion processes for event sequence forecasting," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[26] Y. Yang, M. Uy, and A. Huang, "FinBERT: A pretrained language model for financial communications," *ArXiv preprint*, vol. arXiv:2006.08097, 2020.

[27] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha, "Transformer hawkes process," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[28] X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing, "Dags with no tears: continuous optimization for structure learning," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 9492–9503.

[29] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 6572–6583.