IMPERIAL

Causal Attention-Based Graph Neural Networks for Earnings Momentum Forecasting with LLM-Augmented Temporal Knowledge Graphs

Author:

James Bell-Thomas

Email: james.bell-thomas24@imperial.ac.uk

CID: 06006794

Imperial College London
Advanced Computing MSc
Department of Computing

Submitted on 10th September 2025

Acknowledgements

I would like to express my sincere gratitude to my academic supervisor, Ovidiu Serban, for his continued support and guidance throughout this project. I also thank the Imperial College London Data Science Institute for providing access to their computational resources, which were essential for carrying out this research.

Declaration

The accompanying research project report entitled: "Causal Attention-Based Graph Neural Networks for Earnings Momentum Forecasting with LLM-Augmented Temporal Knowledge Graphs" is submitted in partial fulfillment of the requirements for the degree of Master of Science in Advanced Computing at Imperial College London. The report is based upon independent work by the candidate. All contributions from others have been acknowledged above. The views expressed within the report are those of the author and not of Imperial College London. The code that implemented the methodology and created the results in this project can be found at: https://github.com/jamiebellthomas/apollo

I hereby declare that the above statements are true.

Signed (author)

.....

Full Name

.....

Date

.....

Executive Summary

Earnings momentum, most prominently observed through post-earnings announcement drift (PEAD), is one of the most studied yet persistent anomalies in financial markets. Following an earnings surprise, stock prices often continue to drift in the same direction for several weeks or months, contradicting the efficient market hypothesis. While traditional factor models and econometric approaches have attempted to capture this phenomenon, they fail to explain the structured, causal, and temporal mechanisms underlying the persistence of drift. Recent work has explored machine learning models, but these approaches either ignore temporal dynamics or struggle to integrate heterogeneous sources of financial information effectively.

This thesis proposes a novel framework that combines large language models (LLMs), temporal knowledge graphs (KGs), and causal attention-based graph neural networks (GNNs) to forecast earnings momentum. The methodology begins by collapsing unstructured financial text — including SEC filings and news articles — into high-information-density facts enriched with attributes such as sentiment, event type, and precise timestamps. These facts, alongside quantitative company indicators, are used to construct snapshot KGs centred on each earnings announcement. The design ensures strict temporal causality: only information available before the announcement is included, and edges are weighted with temporal decay functions to capture diminishing relevance of older events.

Five progressively more advanced GNN architectures were developed and evaluated. Model 1 employed a scalar temporal encoding, while Model 2 introduced Time2Vec for richer temporal representation. Model 3 removed temporal encoding as a baseline. Models 4 and 5 introduced attention mechanisms, with Model 5 incorporating optimisations such as edge-aware weighting, entropy sparsity, and pre-gating. To address instability across random seeds, an ensemble evaluation strategy was used: thirty models per architecture were trained, and predictions were aggregated through majority voting, yielding more stable and interpretable results.

Results demonstrate a clear progression across model iterations. Model 3, which lacked temporal encoding, performed worst, confirming the necessity of temporal representation. Models 4 and 5, both attention-based, significantly outperformed earlier architectures. Model 5 achieved the strongest overall performance, with an AUC of 0.642, high weighted precision, and improved stability in ensemble diagnostics. Importantly, while class 1 recall remained low, precision was consistently high, meaning that when the model identified a case as positive it was highly likely to be a true case of earnings momentum. This precision-heavy bias is particularly desirable in financial contexts, where minimising false positives is critical for actionable predictions.

Explainability analysis was conducted using attention weightings, aggregated by event type clusters and temporal windows. The analysis revealed that negative classifications were strongly associated with clusters linked to risk, restructuring, and regulation, while positive classifications drew more heavily on growth-oriented and market expansion events. Temporal attention analysis showed that the model systematically down-weighted older information, with strong emphasis on facts within two months prior to announcements. Examination of false negatives revealed that samples unanimously misclassified by the ensemble tended to contain events from clusters most strongly associated with negative classifications, explaining the systematic difficulty in recovering recall.

The contributions of this thesis are fourfold. First, it introduces a novel KG construction pipeline that collapses financial text into high-density fact representations enriched with sentiment, temporal, and categorical attributes. Second, it proposes a hybrid labelling scheme that combines EPS surprises with medium-term cumulative abnormal return trajectories to isolate genuine PEAD cases. Third, it develops and evaluates five GNN architectures, culminating in an optimised attention-based model that outperforms both primitive baselines and state-of-the-art methods such as XGBoost-based predictors and textual sentiment models. Finally, it presents an interpretability framework based on attention weightings, enabling the identification of event types and time windows most influential to predictions.

Overall, this work serves as a promising proof of concept. It demonstrates that causal, attention-based GNNs operating on LLM-augmented temporal knowledge graphs can provide accurate, interpretable, and financially meaningful forecasts of earnings momentum. Although limitations remain — particularly the scarcity of proprietary financial data and the resulting low recall — the framework establishes a strong foundation for future work. Expanding the range of quantitative indicators and accessing richer news datasets will further enhance the predictive power and generalisability of this approach.

Contents

Figures

List of Tables

1. Introduction

1.1 Earnings in Financial Markets

1.1.1 Definition and Role of Earnings Per Share (EPS)

Earnings per share (EPS) is one of the most widely used indicators of a company's profitability and financial health. It represents the portion of a firm's net earnings that is attributed to each outstanding share of common stock, serving as a measure of the company's ability to generate value for shareholders.

$$\text{EPS} = \frac{\text{Net Income} - \text{Preferred Dividends}}{\text{Weighted Average Shares Outstanding}} \tag{1.1}$$

Preferred dividends are fixed payments owed to holders of preferred shares, a class of stock that receives dividends before common shareholders. Since these payments are contractually guaranteed, they are subtracted from net income when calculating earnings per share (EPS), ensuring that EPS reflects only profits available to common shareholders. Weighted average shares outstanding adjusts for changes in share count over the reporting period by weighting each amount according to how long it was in effect. This provides a fairer basis for EPS, especially when issuance or repurchases occur [1].

EPS is disclosed in quarterly earnings reports, typically 10-Q (quarterly) and 10-K (annual) filings in the United States [2]. Investors and analysts closely monitor EPS as a standardised indicator of profitability per share, which facilitates comparisons across firms regardless of size or capital structure. For instance, two technology firms with very different revenues can be directly compared on the basis of EPS, enabling assessment of relative efficiency and profitability.

When reported EPS diverges significantly from analyst expectations—an earnings surprise—the market often reacts sharply. Such reactions reflect reassessments of growth prospects, competitiveness, and management credibility, and are amplified by media coverage and institutional repositioning. These dynamics frequently lead to momentum effects that this project seeks to forecast [3].

1.1.2 Analyst Forecasts and Earnings Expectations

In the lead-up to a company's quarterly earnings release, a key benchmark used by market participants is the analyst earnings forecast. These forecasts are produced by equity research analysts, typically employed by investment banks or financial services firms, who use a combination of historical data, industry trends, company guidance, and macroeconomic indicators to estimate a firm's earnings per share (EPS) for the upcoming period [4].

Once individual forecasts are submitted, financial data providers such as Refinitiv, FactSet, or Bloomberg aggregate them into what is known as the consensus estimate. This figure represents the average or median prediction of EPS from a wide sample of analysts and is treated by the market as the de facto expectation [5]. Consensus estimates are updated continuously as new information becomes available or as analysts revise their models.

These estimates serve a critical role in market functioning. Rather than evaluating a company's earnings report purely on its own merits, investors interpret the results relative to the consensus. A company that reports EPS above the consensus is said to have delivered a positive earnings surprise, while one that falls short has delivered a negative surprise. The market typically prices in the expectation ahead of time, meaning that the surprise component — the delta between actual and expected EPS — becomes the new information that drives price adjustments upon release [6].

The existence of a widely shared benchmark like the consensus estimate is part of what enables rapid and coordinated market reactions. Because most institutional investors and automated trading systems track the same benchmark, earnings releases can cause significant and immediate price volatility, even if the difference from expectations is only a few cents per share. The strength of the

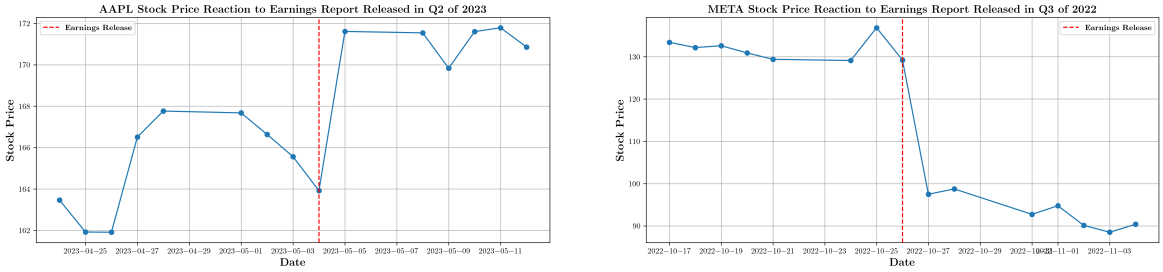
reaction often depends not only on the size of the surprise but also on the broader context, such as the company's guidance, sector performance, and prevailing macroeconomic sentiment.

1.1.3 Market Sensitivity to Earnings Announcements

Quarterly earnings announcements are among the most impactful events in financial markets. When a company publishes its results, the market rapidly adjusts its valuation in response to how the reported earnings compare to expectations. This adjustment is often immediate and significant, especially when the reported EPS differs from the consensus forecast. These reactions are clearly visible in intraday and daily price charts. A typical pattern is a price gap between the previous close and the next open, reflecting a sudden revaluation of the firm's outlook. Positive surprises tend to produce sharp upward movements, often followed by continued momentum. Negative surprises frequently lead to abrupt declines, which may be followed by a slower recovery or continued underperformance. The figures below present real-world examples of this behaviour, demonstrating the sensitivity of equity prices to earnings announcements.

Case Study #1: Apple Inc. (AAPL) Q2 2023 Earnings Beat AAPL stock exhibited a 4.5% price increase in the trading session following its Q2 2023 earnings release (Figure 1.1a), which beat consensus EPS estimates by 10%.

Case Study #2: Meta Platforms (META) Q3 2022 Earnings Miss META stock dropped roughly 25% immediately following a Q3 2022 earnings miss (Figure 1.1b), driven by disappointing revenue guidance and a larger-than-expected decline in ad sales.



(A) AAPL stock price movements at the end of Q2 2023.

(B) META stock price movements at the end of Q3 2022.

FIGURE 1.1: Comparison of stock price movements following earnings announcements for AAPL and META.

1.2 Understanding Earnings Momentum

1.2.1 Earnings Surprises and Price Impact

An earnings surprise occurs when a company's reported earnings per share (EPS) diverges from the consensus estimate of financial analysts. Surprises are classified as positive when reported EPS exceeds expectations, and negative when it falls short. Even small surprises can trigger substantial price adjustments by altering market expectations.

The market response is usually swift, as shown in Figures 1.1a and 1.1b. Prices often move sharply within minutes of the release, driven by institutional investors, retail traders, and automated systems. Positive surprises tend to lift share prices, while negative ones trigger declines, particularly if they damage confidence in management or hint at structural weaknesses.

The reaction also depends on context. A modest beat paired with raised guidance can spark strong gains, whereas a large beat followed by cautious commentary on costs or demand may dampen enthusiasm. This shows that interpretation depends not just on the reported numbers but also on the accompanying narrative and outlook.

This project does not aim to predict surprises directly but treats them as the building blocks of earnings momentum.

1.2.2 The Life Cycle of Earnings Momentum

Understanding the dynamics of earnings momentum requires examining the sequence of events that unfold following an earnings announcement. This sequence, often referred to as the "earnings momentum life cycle," encapsulates how information is disseminated, interpreted, and acted upon in financial markets. The foundational work by Bernard and Thomas [7, 8] provides critical insights into this process, particularly through their documentation of the post-earnings announcement drift (PEAD) phenomenon.

Surprise Event

The cycle begins with an earnings announcement that deviates from market expectations. An earnings surprise occurs when a company's reported earnings per share (EPS) differ from the consensus forecasts made by analysts. Such surprises can be positive (exceeding expectations) or negative (falling short). Bernard and Thomas [7] highlighted that these surprises are not fully absorbed by the market instantaneously, setting the stage for subsequent price adjustments.

Immediate Price Reaction

Upon the announcement, the market reacts quickly, adjusting the company's stock price to reflect the new information. However, this immediate reaction often underrepresents the full implications of the earnings surprise. The initial price movement captures only a portion of the total adjustment needed to align the stock's valuation with the company's revised earnings outlook.

Analyst Revisions and Underreaction

Following the initial reaction, analysts and investors reassess their projections and valuations. Bernard and Thomas [8] observed that investors tend to underreact to earnings surprises, failing to fully incorporate the implications of current earnings for future performance. This underreaction leads to a gradual adjustment process, where subsequent earnings forecasts and stock valuations are incrementally revised over time.

Momentum Persistence

The underreaction contributes to a persistence in stock price movements in the direction of the earnings surprise—a phenomenon known as PEAD. Bernard and Thomas [8] found that this drift can continue for up to 60 trading days post-announcement, with a significant portion of the drift occurring around subsequent earnings announcements, as seen in Figure 1.2. This persistence suggests that the market continues to assimilate the information from the initial surprise over an extended period, leading to sustained momentum in stock prices.

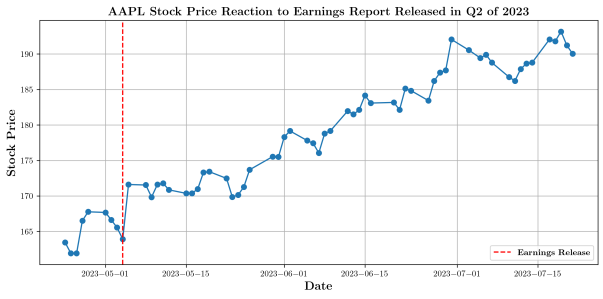


FIGURE 1.2: PEAD demonstration – AAPL stock price movements 60 days after the end of Q2 2023.

1.2.3 Empirical Evidence of Earnings Momentum

Earnings momentum is one of the most well-documented inefficiencies in financial markets. Bernard and Thomas [7, 8] showed that firms reporting positive earnings surprises tend to experience continued stock price increases for several weeks or months after the announcement. This effect, known as post-earnings announcement drift (PEAD), reflects a delayed market reaction and cannot be fully explained by risk or information timing.

Subsequent studies have reinforced these findings. Jegadeesh and Livnat (2006) demonstrated that the drift is strongest when firms beat both earnings and revenue expectations, suggesting that markets react more strongly when multiple financial signals align. Chan, Jegadeesh, and Lakonishok [9] showed that the effect persists even after accounting for firm size, book-to-market ratio, and past returns, indicating that momentum is not confined to niche asset classes.

The behavioural explanation for this drift is gradual information diffusion. Hong and Stein [10] proposed that different investor groups update their expectations at different speeds, leading to a slow and persistent adjustment in prices. This underreaction gives rise to exploitable patterns, particularly when companies repeatedly exceed expectations over multiple quarters.

1.3 The Importance of Anticipating Momentum

1.3.1 Investor Incentives for Forecasting Earnings Momentum

Investors have strong incentives to identify firms in earnings momentum regimes before these patterns are fully recognised by the market. Firms that consistently beat analyst expectations often reflect underlying operational strength or improving fundamentals. However, due to behavioural underreaction, stock prices tend to adjust slowly, creating a window in which investors can act on the information before it is fully priced in. Being able to forecast momentum regimes in advance allows for more informed capital allocation. Rather than reacting to individual earnings surprises, which are noisy and often ambiguous, a model that captures sustained patterns of outperformance offers a more reliable basis for medium-term investment decisions. Additionally, momentum signals that are grounded in observable causes — such as product launches, macroeconomic shifts, or analyst behaviour — are more likely to be adopted by institutional investors. These actors typically favour models that provide not just predictive accuracy, but also transparency and economic justification.

1.3.2 Why Existing Models Fall Short

Despite the long-standing documentation of earnings momentum, most existing models fail to capture the structured, causal nature of the phenomenon. Traditional quantitative approaches, such as trend-following and time-series momentum strategies, typically rely on historical price patterns. While effective during strong directional markets, these models perform poorly during periods of volatility or when momentum regimes break down [11].

Factor-based models like the Fama-French three-factor and five-factor frameworks explain stock returns using size, value, profitability, and investment metrics, but notably exclude momentum. The momentum anomaly, first formalised by Jegadeesh and Titman [12], remains one of the strongest empirical predictors of returns. Its omission represents a fundamental limitation of such models in capturing regime-level investor underreaction.

Recent deep learning approaches, including LSTMs and transformers, have been applied to financial time series prediction. While these models can learn complex nonlinear relationships, they often lack transparency and are difficult to interpret. This black-box nature makes them poorly suited to contexts that require explainable decision-making, such as institutional investing [13]. Graph-based methods, meanwhile, have shown promise in modelling financial entities and their relationships. However, most current approaches rely on static or per-snapshot graphs, lacking mechanisms for handling temporal evolution or dynamic event-driven updates. As surveyed by Chen et al. [14], the integration of time-aware reasoning and persistent memory remains an open challenge in financial graph learning.

1.3.3 The Case for Causal and Temporal Reasoning

Building reliable models of earnings momentum requires representing how financial signals evolve and interact over time, rather than treating them as static or independent inputs. Momentum arises not from isolated announcements, but from sequences of events whose order, timing, and orientation determine how investors update expectations. Models that fail to account for this dynamic structure risk overemphasising outdated signals or missing the context in which they occur.

Temporal reasoning is crucial because the relevance of events decays with age, and the order in which they occur shapes their interpretation. A negative earnings miss that follows repeated analyst downgrades may reinforce bearish sentiment, whereas the same miss after a period of upgrades could be interpreted as a temporary correction. Representations that explicitly encode timestamps and recency enable models to prioritise recent and sequentially significant events, which has been shown to improve predictive accuracy in dynamic domains [15, 16].

Sentiment further enriches this representation. Events such as acquisitions, regulatory penalties, or product launches differ not only in type but also in directional impact. Embedding sentiment as a continuous or categorical attribute allows models to distinguish between positive and negative influences on firm valuation, aligning more closely with how investors process news. Recent advances in financial natural language processing highlight that sentiment-aware modelling yields measurable gains in prediction tasks [17].

An event-centric knowledge graph that integrates both temporal decay and sentiment thus offers a principled foundation for momentum prediction. By treating events as fact nodes enriched with time and sentiment attributes, and linking them to companies and related events, the resulting structure captures both evolving market narratives and their directional effects. This design directly addresses the shortcomings of flat, correlation-driven models by embedding both time and sentiment into the reasoning process, producing a representation more consistent with how information propagates in financial markets.

1.4 Motivation for a New Approach

1.4.1 The Role of Unstructured Data and News in Momentum

While earnings momentum is traditionally quantified using structured financial indicators such as EPS and analyst forecasts, unstructured information sources often act as early signals of momentum shifts. Financial news, earnings call transcripts, and SEC filings contain rich qualitative information, including sentiment, forward-looking statements, and contextual detail, which may precede observable changes in fundamentals. These signals are particularly valuable when analyst expectations lag behind evolving narratives or shifts in market perception. For example, media coverage of a product launch or leadership change can influence investor sentiment in advance of earnings updates, effectively shaping market expectations and price movement. Empirical studies have shown that textual information extracted from financial news can enhance forecasting accuracy when used alongside conventional numerical indicators [18]. This supports the view that unstructured data is not merely descriptive, but can encode latent signals indicative of future firm performance and, by extension, future momentum.

1.4.2 LLMs for Extracting Structured Financial Knowledge

Transforming unstructured financial text into structured representations is essential for downstream modelling tasks such as graph construction, temporal reasoning, and classification. Traditional information extraction pipelines — typically built around rule-based systems or syntactic parsers — often suffer from brittleness and limited coverage, particularly in the nuanced language of financial news and disclosures.

Large Language Models (LLMs) now offer a compelling alternative. They can extract entities, relations, and events from raw text with significantly higher adaptability and semantic awareness. Given a news paragraph or an earnings report, an LLM can identify company names, actions (such as acquisitions or earnings downgrades), affected parties, and even assign sentiment scores to the event. This allows for the construction of structured representations such as knowledge graphs, which preserve the semantics and chronology of real-world developments.

Recent work has demonstrated that LLM-based extraction systems, including domain-specific models like FinBERT, outperform traditional approaches in both accuracy and financial relevance [19]. By leveraging pre-trained financial-domain LLMs or adapting general-purpose ones through

prompting or fine-tuning, we can extract high-quality structured information from noisy and diverse financial text sources at scale.

1.4.3 Why Use Temporal, Sentiment-Aware Knowledge Graphs

Knowledge graphs offer a principled way to integrate heterogeneous and evolving information, which is essential for modelling the financial events that underpin earnings momentum. A static or purely company-centric representation cannot capture how market narratives evolve. An effective representation must encode both temporal structure and the overall sentiment of events.

An event-centric graph addresses this need by treating each extracted fact as a node with attributes such as timestamp, sentiment score, event type, and the set of related companies. Edges encode temporal adjacency, semantic relatedness, and company associations, so the graph reflects how discrete facts accumulate into narratives that move expectations. This design directly supports temporal reasoning, where recency and event order shape interpretation [15, 16, 20].

Polarity-aware modelling further strengthens the representation by embedding sentiment as a continuous attribute on nodes or edges. Market reactions are asymmetric: positive announcements such as earnings beats, partnerships, or successful product launches can have compounding effects, while negative news such as litigation or downgrades can produce sharper declines. Incorporating continuous sentiment into graph learning helps models distinguish otherwise similar event types with opposing implications, and has been shown to improve financial prediction when combined with event structure [17, 21].

Event-centric temporal graphs also align with practical data pipelines. LLMs can extract fact-level events from noisy text, attach timestamps and tickers, and update the graph as new information arrives. Finance-specific resources demonstrate the feasibility of such pipelines at scale, for example FR2KG for automated financial KG construction, which shows how structured event knowledge can be assembled from unstructured reports [22]. Taken together, a temporally evolving, sentiment-aware event graph provides a representation that is well matched to how information propagates in markets and therefore forms a suitable foundation for anticipating earnings momentum.

1.5 Overview of the Proposed Methodology

1.5.1 Fact-Level Event Extraction with LLMs

A central challenge in building an event-centric knowledge graph is the transformation of raw financial news and filings into structured representations that can be used for downstream reasoning. Full articles often contain a mixture of relevant facts, background detail, and noise, making direct modelling ineffective. To address this, the first stage of the methodology focuses on extracting atomic, fact-level events from unstructured text using large language models (LLMs).

Each fact is represented as a node in the graph, enriched with attributes that capture its essential characteristics. These include the event timestamp, the set of companies involved, the type of event (for example earnings release, product launch, or regulatory action), and a sentiment score that summarises the impact implied by the text. By breaking articles down into discrete facts, the approach avoids overgeneralisation and ensures that fine-grained signals are preserved.

For example, consider the news sentence: *"On 12 March 2023, Company A announced quarterly earnings that exceeded analyst expectations."*
From this, the extraction stage would produce a single fact node with attributes:

- Date: 12 March 2023
- Companies: Company A
- Event type: Earnings release

- Sentiment score: +0.8 (derived from *"exceeded expectations"*)
- Text summary: *"Company A exceeded predicted quarterly earnings"*

This fact is then integrated into the graph, where it can be linked temporally and semantically to related companies.

The extraction process is designed to balance accuracy with scalability. LLMs are used to identify and segment factual statements within an article, filtering out rhetorical or contextual information. Each fact is then standardised into a structured schema that can be consistently integrated into the knowledge graph. This schema provides the flexibility to capture both categorical information, such as event type, and continuous values, such as sentiment intensity.

The outcome of this stage is a stream of atomic event-facts, each linked to their publication date and associated companies. These fact nodes form the foundation of the graph, enabling temporal and sentiment-aware reasoning in subsequent stages. By focusing on fact-level extraction rather than whole-article representation, the methodology reduces noise, improves interpretability, and ensures that the knowledge graph more faithfully reflects the actual sequence of events shaping market dynamics.

1.5.2 KG Construction and Temporal Updates

After fact-level extraction, graphs are assembled only at the point of inference. Rather than maintaining a global, ever-growing structure, the system builds a snapshot knowledge graph for each earnings announcement. The snapshot captures the information environment immediately prior to the release and is used for both training and prediction.

Each snapshot is centred on a single focal company node that will report earnings. The snapshot includes every fact node derived from news or filings that mention the focal company within a configurable look-back window. The look-back is a hyperparameter, with an upper bound of one quarter. Facts outside this window are excluded from the snapshot.

There is a single edge type. Company–fact edges connect the focal company to all facts that mention it within the window. If a fact also mentions other companies, edges connect that same fact node to those additional company nodes. There are no fact–fact edges and no company–company edges.

Attributes are assigned as follows. Company nodes are instantiated fresh for each snapshot and carry quantitative features that are valid at the snapshot time, for example recent returns, valuation ratios, or analyst consensus fields. Fact nodes carry the event classification and a short raw-text summary produced by the extractor. Company–fact edges carry continuous sentiment and a time-decay weight that reflects recency within the window. Decay is applied on the edge so that older facts within the same snapshot naturally receive lower effective weight than recent ones, without imposing an explicit temporal chain among facts.

This snapshot design provides three benefits. First, it guarantees that only information available at inference time enters the model, which prevents leakage. Second, it reduces computational load. Tracking a large, evolving global graph and repeatedly extracting per-company subgraphs would be memory and runtime intensive. Constructing compact snapshots on demand keeps graphs small, regular, and aligned with the prediction task. Third, it improves feature consistency. Company nodes are rebuilt per snapshot with contemporaneous quantitative metrics, and edges encode sentiment and decay in a uniform way across companies and dates.

Illustrative structure at inference time A central company node is connected to all facts in the look-back window that reference it. Each of those facts connects to any other companies it mentions, which are instantiated within the same snapshot and can contribute contextual signal via their edges from the shared facts. The result is a hub-and-spoke event graph focused on the reporting company, enriched with sentiment scores and temporal information embedded within edges and quantitative attributes on company nodes, and constructed strictly from information available before the announcement.

1.5.3 Subgraph-Based GNN Classification of Earnings Momentum

With snapshot graphs assembled, the prediction task is formulated as a binary classification problem: the goal is to determine whether the focal company will experience positive earnings momentum following its announcement. To train such a classifier, labels must be constructed in a way that reflects genuine market reactions. Each snapshot is therefore assigned a binary outcome based on the combination of:

1. Whether the company delivered a positive or negative earnings surprise relative to analyst expectations
2. Whether the stock price displayed evidence of post-earnings announcement drift (PEAD) consistent with that surprise.

This label construction ensures that the task captures both the earnings event itself and the subsequent market adjustment, rather than short-lived or noisy movements.

Once labelled, each snapshot graph becomes a self-contained training, validation, or testing instance. This creates a dataset of graph samples aligned with discrete earnings events, allowing the model to learn from historical periods and be evaluated on unseen announcements.

To process these graphs, graph neural networks (GNNs) are applied to propagate information along company–fact edges. Through message passing, signals from event nodes are aggregated into company representations, and related firms mentioned in the same facts contribute contextual influence. The focal company node thus accumulates a representation that reflects both its quantitative metrics and the informational environment of recent news. Because fact nodes are derived from text, they are first vectorised into embeddings—for example using pre-trained language models—so that textual event descriptions can be integrated numerically into the graph.

The final embedding of the focal company node is fed into a classification layer that predicts the binary momentum outcome. This formulation directly aligns the model's inputs and outputs with the practical forecasting problem faced by investors: given all available information prior to an earnings release, predict whether the firm is likely to exhibit sustained positive or negative momentum in its post-announcement returns.

1.5.4 Explainability via Attribution

An important component of the methodology is to provide transparency into how the model arrives at its predictions. Since the input to the classifier is a heterogeneous graph combining financial metrics, text-derived facts, and sentiment-weighted edges, it is essential to identify which features and structures are most influential in driving the outcome.

Explainability is approached through attribution methods applied to the trained graph neural network. Rather than treating the model as a black box, attribution assigns importance scores to nodes, edges, and attributes within each snapshot. This allows us to highlight which elements of the input graph were most critical to the classification of earnings momentum. In practice, the focus is on identifying:

- Event types that consistently influence predictions, such as earnings releases, analyst reports, or regulatory actions.
- Sentiment signals on company–fact edges, to show whether the model relies more heavily on positive or negative sentiment scores when making its decision.
- Temporal characteristics, by evaluating whether more recent facts within the look-back window contribute disproportionately compared to older ones.

The outcome of attribution analysis is twofold. First, it provides assurance that the model is learning from meaningful financial signals rather than noise. Second, it offers interpretability for analysts and practitioners by clarifying which categories of information tend to drive momentum predictions. In this way, attribution bridges the gap between predictive accuracy and financial insight, ensuring that the system remains both effective and explainable.

1.6 Research Objectives and Contributions

This project aims to investigate whether a structured, temporally evolving financial knowledge graph, enriched with causal and sentiment-aware information, can enhance the prediction of earnings momentum. It also explores how graph-based representations can improve interpretability and forecasting transparency in financial machine learning systems.

1.6.1 Key Research Questions

This project is guided by the following key questions:

- Can earnings momentum be effectively predicted using a snapshot-based knowledge graph constructed from unstructured qualitative financial text and quantitative financial data?
- Does incorporating sentiment and temporal decay improve the graph's ability to capture financially meaningful signals?
- Can a graph neural network operating on snapshot subgraphs centred on earnings announcements learn predictive patterns from structured financial contexts?
- Does the use of fact-level events extracted by large language models provide a richer signal than traditional news sentiment scores?
- To what extent can attribution help interpret the model's predictions, by identifying which event types, sentiment orientations, and recency windows were most influential?

1.6.2 Novel Contributions of This Project

This project makes the following key contributions:

- A snapshot-based event-centric financial KG design that constructs subgraphs only at earnings announcement dates. Unlike prior work on global financial knowledge graphs or continuous temporal graphs, this approach avoids the complexity of evolving structures and directly aligns the graph with the point of inference, providing a new computationally efficient way of modelling pre-announcement information.
- A joint labelling scheme for earnings momentum that combines earnings surprise with evidence of post-earnings announcement drift (PEAD). Existing studies typically treat surprise and drift separately; this work proposes a unified definition of momentum outcomes that reflects both the trigger (EPS shock) and its persistence in the market response.
- Integration of heterogeneous signals within an event-centric graph where quantitative firm metrics, vectorised text-based facts, and sentiment-weighted time-decayed edges are combined in a single GNN framework. While prior models incorporate subsets of these signals in isolation, this project advances a unified architecture that embeds them together in a structured, relational representation.
- An attribution-based explainability module tailored to earnings prediction, designed to identify which event types, sentiment polarities, and recency windows most strongly influenced model outputs. This moves beyond generic feature attribution by focusing specifically on categories of information that matter in financial forecasting contexts.

2. Literature Review

2.1 Financial Forecasting from Unstructured Data

2.1.1 Textual Signals in Finance

A growing body of research has demonstrated that financial markets respond not only to structured numerical indicators but also to qualitative information found in natural language sources. Corporate disclosures, financial news, analyst commentary, and even social media collectively shape investor sentiment and expectations, often in ways that precede changes in price or volatility. The idea that text contains predictive signals is now well established, and this insight has led to a wave of text-driven approaches in financial forecasting.

Early foundational studies showed that textual content could reflect or even anticipate economic fundamentals. Tetlock famously demonstrated that negative tone in news articles about financial markets predicted downward price pressure in the short term, highlighting the informational inefficiency of investor responses to qualitative inputs [23]. Similarly, Loughran and McDonald developed a financial-domain sentiment dictionary to better capture how tone in earnings releases and filings correlates with market reactions [24].

Later studies expanded the scope of textual signals beyond tone. Kogan et al. showed that firm-specific language in SEC filings contains information about future returns and volatility [25], while Ding et al. used event-centric representations of financial text to predict price movements with improved granularity [26].

Collectively, this body of work establishes that financial text encodes latent signals about investor expectations, firm trajectories, and market response dynamics. However, most existing approaches rely on document-level sentiment or keyword frequencies, reducing rich, relational information into flat features. As such, they do not account for structured interdependencies between events, actors, and time. This gap motivates the use of knowledge graphs in the present work.

2.1.2 Event-Driven Prediction and News Analytics

Beyond general sentiment, financial prediction research has increasingly focused on event-driven models that aim to extract and exploit discrete occurrences—such as earnings reports, mergers, downgrades, and lawsuits—embedded in financial news and filings. These events are often viewed as more stable and interpretable units of information than raw sentiment scores, providing clearer causal signals that link news to subsequent market behaviour.

Ding et al. proposed one of the first deep learning-based pipelines for financial event extraction and price movement prediction. They introduced a structured representation of events using tuples of the form (actor, action, object) and used convolutional and dependency-based neural networks to encode them for downstream stock prediction tasks [27]. This model demonstrated that event representations outperformed traditional bag-of-words and sentiment-based approaches in forecasting short-term price direction.

Subsequent work refined this idea by incorporating event context, co-occurrence patterns, and temporal dependencies. Hu et al. developed a hierarchical attention model that considered both event type and argument structure to determine an event's importance to the target stock [28].

These approaches have established the viability of using structured event representations extracted from text for financial forecasting. However, they often rely on pre-defined event taxonomies or limited relation types, and they rarely capture sentiment, causal structure, or temporal decay explicitly. Moreover, most existing systems treat events as isolated or independent signals, rather than linking them within an evolving knowledge structure. This motivates the use of knowledge graphs as a more flexible and expressive representation, capable of capturing not only event content but also the relational dynamics between events, entities, and outcomes over time.

2.1.3 Sentiment, Tone, and Predictive Performance

Sentiment analysis has long been a core technique in financial text mining, used to infer market sentiment, analyst tone, or managerial optimism from qualitative disclosures. It offers a scalable way to transform unstructured text into numerical indicators for downstream forecasting tasks, particularly in the absence of explicit event annotations.

Early work in this area applied general-purpose sentiment lexicons to financial news and filings. However, these tools often misinterpreted domain-specific language. For example, Loughran and McDonald developed a finance-specific sentiment dictionary after showing that traditional tools frequently misclassified neutral financial terms like "liability" or "capital" as negative [24]. Their dictionary, tailored for 10-K filings, became a standard in financial sentiment analysis.

Subsequent approaches leveraged supervised machine learning and deep learning to infer sentiment more flexibly. Bollen et al. demonstrated that sentiment extracted from Twitter could predict broad market movements such as the DJIA index [29]. More recent models like FinBERT, a domain-adapted version of BERT trained on financial corpora, have improved accuracy in detecting sentiment in earnings calls, press releases, and analyst reports [17]. These tools enable sentence-level or clause-level sentiment evaluation using context-sensitive embeddings.

Despite these advances, sentiment remains limited in several ways. Most approaches assign sentiment at the document or sentence level, discarding relationships between entities or events. Furthermore, sentiment signals are usually treated as isolated time-series inputs rather than embedded within a structured reasoning framework. Very few systems incorporate sentiment into graph-based models, where it can be attached directly to specific events and their links to companies.

In this project, sentiment is represented as an attribute of fact–company relations within a knowledge graph. This design ensures that sentiment is preserved in context, reflecting not only the tone of a disclosure but also its connection to specific firms and events. By embedding sentiment directly into structured representations, the model captures both the qualitative nuance of financial language and its relevance to downstream tasks such as predicting post-earnings announcement drift.

2.2 Financial Knowledge Graph Construction

2.2.1 Traditional and ML-Based KG Construction

The construction of financial knowledge graphs (KGs) has evolved significantly over time, transitioning from rule-based systems to advanced machine learning (ML) and deep learning techniques. Early pipelines were dominated by rule-based natural language processing (NLP), relying on grammars, lexicons, and dependency parsers to extract subject–predicate–object triples from structured financial sources such as 10-K filings, earnings reports, and news articles. These approaches, while precise within narrow domains, were brittle to linguistic variation and lacked generalisability. One foundational framework in this space was Open Information Extraction (OpenIE), which aimed to extract relational triples without predefining schemas, but often produced overly generic or noisy outputs [30].

To improve robustness and scalability, statistical ML techniques such as Conditional Random Fields (CRFs) and Support Vector Machines (SVMs) were adopted for named entity recognition (NER) and relation extraction. These models learned from annotated examples but still relied heavily on hand-crafted features and performed poorly on complex financial texts [31]. Moreover, the scarcity of high-quality annotated data in the financial domain limited their practical deployment.

The advent of deep learning, particularly transformer-based architectures, revolutionised the field. Pretrained language models like BERT, and its domain-specific variants such as FinBERT, enabled more accurate and context-aware extraction of financial entities and relations. FinBERT was initially developed for financial sentiment classification [17], but has since been fine-tuned for tasks like entity and relation extraction with considerable success [32].

To standardise evaluation and foster progress in financial IE, recent benchmarks such as FIRE (Financial Information Relation Extraction) have been proposed. FIRE provides sentence-level annotations of financial texts with labelled entity pairs and relation types, supporting robust model development for relation extraction in financial news and reports [33]. This enables a more systematic comparison of models across tasks and datasets.

Despite these advances, most existing financial KGs remain flat and static, capturing isolated document-level triples without modelling temporal order or sentiment intensity. To overcome these limitations, the present work employs an event-centric, snapshot-based KG design. Unstructured text is decomposed into atomic fact nodes with attributes including event classification and short raw-text summaries. Companies are instantiated as nodes within each snapshot, carrying contemporaneous quantitative metrics. The only edges in the graph link companies to fact nodes, and these edges are annotated with continuous sentiment scores and time-decay weights that down-weight older information within a configurable pre-announcement window. Rather than maintaining a global evolving graph, compact snapshot subgraphs are constructed around the company reporting earnings, incorporating all facts that mention it within the look-back window and linking any other companies referenced in those facts. This representation is temporally grounded, computationally tractable, and provides the structured context for subsequent temporal encoding and learning.

2.2.2 Edge Weighting with Sentiment and Temporal Information

Traditional knowledge graph (KG) construction focuses on extracting static triples (subject–relation–object) from text. However, financial reasoning often demands representations that incorporate both sentiment and temporal evolution. Encoding these dimensions into KGs is essential for modeling the directionality of financial events and tracking how firm-level relationships evolve over time.

Sentiment refers to the evaluative tone or orientation of an event or relationship, ranging from strongly negative to strongly positive. In financial contexts, this includes whether an analyst report conveys optimism or pessimism, whether earnings guidance signals confidence or concern, or whether a regulatory decision is perceived as beneficial or detrimental to a firm. Prior work has explored incorporating sentiment into knowledge graphs by annotating edges with sentiment scores or by using sentiment as a relational modifier. For example, Ding et al. proposed a deep learning method for event-driven stock prediction, where events extracted from news text were represented as dense vectors encoding sentiment intensity and its market impact [26]. Despite these advances, many systems continue to treat sentiment as a surface-level textual feature rather than embedding it directly into the structural semantics of the graph.

Temporal encoding ensures that KGs reflect the evolution of relationships and events over time. This can be achieved by timestamping edges, constructing graph snapshots at discrete time intervals, or learning continuous time-aware embeddings. Foundational models such as Know-Evolve [15] and DyRep [34] proposed dynamic KGs where entity and relation representations evolve based on historical interactions. More recent models, like the Temporal Graph Network (TGN) [16] and TGN-based financial systems [35], use temporal attention and edge decay to capture the fading relevance of past information—a crucial mechanism for modeling rapidly evolving financial data. In addition to decay-based mechanisms, approaches such as Time2Vec [36] represent temporal information as learnable vector embeddings, capable of capturing both linear and periodic patterns. Incorporating Time2Vec alongside decay allows temporal encoding to model not only the diminishing importance of older events but also cyclical structures, such as quarterly reporting rhythms, that characterise financial markets.

Despite these advances, relatively few systems support sentiment-aware and temporally dynamic reasoning in combination. Many financial KGs remain static or rely on coarse temporal segmentation, and sentiment is often treated as an auxiliary feature rather than being embedded directly into relational structure. To address this gap, the proposed design employs a bipartite snapshot KG linking companies to fact-level events, where each edge is annotated with a continuous sentiment score and a temporal representation. Edge weights are constructed as a composite of sentiment intensity, a scalar decay function that reduces the influence of older events within the pre-announcement

window, and a Time2Vec embedding [36]. This approach embeds both informational intensity and temporal relevance directly into the graph, ensuring that connections reflect not just whether a company is associated with an event, but also how strongly and how recently that event should affect earnings momentum prediction.

2.2.3 Persistent and Evolving Financial KGs

Most existing financial knowledge graphs are static or document-scoped, focusing on extracting facts from individual reports, filings, or articles. However, understanding complex financial behaviour often requires integrating events and relations across time. Persistent and evolving knowledge graphs aim to retain this historical context by continuously updating and integrating information into a dynamic structure.

FinKG is one of the more comprehensive attempts at financial KG construction, combining various relation types and multi-source data into a unified, temporally annotated graph [37]. Similarly, FinDKG introduces a document-level KG approach that preserves discourse-level context from financial texts such as earnings calls, enabling richer downstream tasks like sentiment reasoning [38]. These approaches highlight the importance of retaining temporal and relational structure beyond isolated triples.

While promising, these systems often rely on relatively simple temporal modelling, such as timestamp annotations or coarse update intervals. In addition, many evolving financial KGs are updated at the document level, which can obscure finer-grained event information and make subgraph extraction computationally expensive. As a result, although persistent KGs capture continuity, they can struggle to balance detail with scalability—particularly in tasks like earnings prediction, where the relevant information is naturally bounded to the period leading up to an announcement.

In contrast, the approach in this project does not maintain a persistent or continuously evolving KG. Instead, it constructs compact *snapshot graphs* centred on the focal company at the time of each earnings announcement. These snapshots include all fact nodes extracted from financial texts within a fixed look-back window, linked to the company of interest and any other firms mentioned in those facts. By rebuilding graphs at each decision point, this design avoids the computational overhead of maintaining a global evolving structure while still enabling detailed temporal modelling through edge-level representations of recency and sentiment. The snapshot paradigm therefore provides a tractable yet expressive alternative to persistent financial KGs, delivering the benefits of temporal reasoning without the cost of continuous graph evolution.

2.3 Large Language Models for Financial Information Extraction

2.3.1 Prompt-Based vs Fine-Tuned LLMs

Large Language Models (LLMs) such as GPT-3 and BERT have transformed NLP by enabling models to generalise across tasks with limited supervision. In financial information extraction (IE), two main paradigms dominate: prompt-based inference and fine-tuning.

Prompt-based methods leverage pre-trained models through natural language instructions. Zero-shot and few-shot prompting allow rapid adaptation to new domains with little or no labelled data. GPT-3 and GPT-4, for instance, can be guided to output structured tuples describing events, entities, and relations using prompt templates [39, 40]. This flexibility is attractive in finance, where labelled datasets are scarce, event schemas evolve quickly, and annotation is expensive.

However, prompt-based approaches are highly sensitive to formulation, often inconsistent in format, and prone to hallucination or factual errors [41]. Their outputs typically require post-processing to ensure schema consistency. Fine-tuning offers a more stable alternative when annotated data is available and tasks are well-defined. It adapts model parameters for specific objectives such as Named Entity Recognition (NER), relation extraction, or sentiment classification. Fine-tuned BERT

models on datasets like ACE05 or DocRED achieve state-of-the-art performance across multiple domains, including financial IE [42, 43].

Domain-specific fine-tuning further enhances results. FinBERT, a BERT variant adapted to financial texts, achieves strong sentiment classification performance when trained on the Financial PhraseBank [17]. In contrast, BloombergGPT adopts a different strategy: pretraining from scratch on a 363B-token corpus mixing financial and general-domain data. Despite not being fine-tuned for downstream tasks, it achieves strong zero-shot and few-shot results across financial NER, sentiment, and QA [44]. Together, these models highlight two complementary strategies for domain adaptation: targeted fine-tuning versus large-scale domain-specific pretraining.

Fine-tuned models often provide higher accuracy and stability for fixed tasks but require annotated corpora, costly training pipelines, and regular updates. Prompt-based methods trade peak performance for flexibility, scalability, and ease of deployment, making them attractive in dynamic or low-resource settings [45]. In this project, prompt-based extraction was adopted to enable scalable knowledge graph construction from unstructured text without requiring supervision, ensuring adaptability to evolving financial event types and entity schemas.

2.3.2 Relation and Sentiment Extraction with GPT Models

Large Language Models (LLMs) such as GPT-3 and GPT-4 have introduced powerful capabilities for extracting structured event information from unstructured financial text. With carefully designed prompts, they can identify event boundaries, classify event types, associate events with relevant companies, and assign attributes such as sentiment scores or textual summaries. This reduces dependence on hand-engineered rules and large annotated datasets, which are scarce in the financial domain, while providing adaptability to new contexts and event types.

In finance, where high-quality structured data is limited and markets react to diverse signals, the ability of LLMs to decompose complex text into granular fact nodes is particularly valuable. Instead of producing isolated triples or quintuples, event-centric extraction treats each article, filing, or report as a source of multiple atomic events. These events are then linked to all referenced companies, enabling the construction of bipartite company–fact graphs.

Recent work has demonstrated the feasibility of this approach. For example, Yang et al. (2023) introduced FinGPT, an open-source framework that uses LLMs to extract structured financial information from noisy text [46]. Other studies similarly show that LLMs can perform high-quality extraction without task-specific training, supporting scalable event generation. However, most current systems focus on static or document-level outputs and stop short of building dynamic graph structures aligned with earnings cycles.

The approach in this project extends these ideas by using LLMs not only as extractors of facts but also as enrichers of graph nodes. Each fact node is annotated with an event type, a short textual summary, and a continuous sentiment score derived from context. By converting unstructured text into structured, temporally grounded fact-level units, LLMs provide the raw material for constructing event-centric snapshot knowledge graphs that align closely with the requirements of earnings momentum prediction.

2.4 Temporal and Heterogeneous Graph Learning

2.4.1 Temporal GNNs and Time Decay

In finance and other dynamic domains, graph structure and semantics evolve as new events occur and past information loses relevance. Standard GNNs, which assume static topologies, struggle in such settings. Temporal GNNs address this by incorporating time into node representations and message passing, enabling models to capture both structural evolution and event sequences.

The Temporal Graph Network (TGN) [47] is a foundational model that maintains node memory, updated by new time-stamped interactions via message functions and recurrent units. This allows embeddings to evolve with the arrival of events, making TGN well-suited for forecasting where order and recency matter. TGAT (Temporal Graph Attention Network) [48] instead integrates continuous-time encodings into attention layers, enabling selective weighting of historical neighbours by temporal distance and supporting inductive generalisation to unseen nodes.

Other approaches include DyRep [34], which treats graph evolution as a continuous-time stochastic process. It models both short-term communication and long-term associations with temporal point processes, using decay functions to capture diminishing influence — highly relevant for finance, where recent announcements dominate over older events. CAW (Continuous-time Attributed Walks) [49] takes a different path, replacing memory modules with time-aware random walks, offering efficiency at the expense of explicit node state history.

Together, these architectures highlight the importance of temporal inductive bias in dynamic graphs. Memory-based (TGN), attention-based (TGAT), decay-driven (DyRep), and walk-based (CAW) models each provide distinct ways to encode evolving interactions. In this project, temporal reasoning is applied at the snapshot level: subgraphs are generated per earnings announcement, with edges annotated by decay scalars and learnable time embeddings. This avoids the computational burden of full memory replay while ensuring that recency and cyclical patterns are directly embedded into the graph representation.

2.4.2 Heterogeneous Graph Representation Learning

Early GNN models assumed homogeneous graphs, where all nodes and edges shared the same type. This assumption fails in real-world domains, particularly finance, where entities and relations are diverse. A heterogeneous graph (or heterogeneous information network) contains multiple node types (e.g., companies, people, events) and edge types (e.g., owns, acquires, reports on). Distinguishing these types is essential, as each carries unique semantics.

Financial text illustrates this complexity: a headline such as "Tesla acquires battery startup as Musk announces expansion into energy storage" involves companies, individuals, and industry context. Many heterogeneous KG approaches represent each as a distinct node type linked by relations. In this project, however, information is encapsulated in *fact nodes*, which store attributes such as event type embeddings, sentiment, and text summaries, and link to company nodes. This bipartite structure reduces node-type proliferation while preserving the distinction between fact-level and company-level signals.

Reasoning over such structures depends not only on connections but also on their roles. Without recognising heterogeneity, GNNs may conflate events with different semantics, such as "Tesla announces expansion" vs "Startup acquired Tesla." In bipartite settings, this manifests as treating fact and company nodes identically, eroding representational clarity.

Several architectures address this. Relational Graph Convolutional Networks (R-GCNs) [50] learn separate parameters for each edge type but scale poorly with large relation sets. Heterogeneous Graph Transformer (HGT) [51] introduces type-specific transformations with attention-based aggregation, dynamically weighting neighbours by node and edge type. Heterogeneous Attention Network (HAN) [52] uses metapaths such as "Company $\rightarrow$ Event $\rightarrow$ Person" to capture higher-order semantics, though these require manual design and may not generalise. The principle of type-aware aggregation also applies in bipartite settings, where company and fact nodes fulfil distinct roles.

These models, however, rarely integrate fine-grained sentiment or temporal information, both vital in finance. They are often designed for static or global graphs, whereas this project uses snapshot-based subgraphs aligned with earnings announcements. Within each snapshot, heterogeneity emerges from company nodes with quantitative metrics linked to fact nodes enriched with event and sentiment attributes. Edges further encode sentiment scores and temporal weights. To balance tractability with expressivity, this project adopts a simplified heterogeneous GNN with relation-type-aware aggregation combined with sentiment and temporal encoding.

2.4.3 Subgraph-Based Classification Approaches

Graph Neural Networks (GNNs) have traditionally focused on node- or graph-level tasks, but subgraph-based classification is gaining traction. This paradigm restricts prediction to task-specific regions of the graph, improving interpretability and efficiency when signals are context-dependent.

In financial applications, a firm's state can be represented by a subgraph consisting of the focal company and fact nodes linked within a temporal window. These facts capture events such as acquisitions, executive changes, or peer downgrades, enriched with attributes like event type, sentiment, and text. Other firms mentioned in these events also appear through their connections. Subgraphs in this project are generated fresh at each earnings announcement rather than mined from a persistent KG, ensuring the constructed graph reflects the contemporaneous information environment.

Prior work demonstrates the effectiveness of this approach. GNNExplainer identifies compact subgraphs most relevant to predictions [53]; Liu et al. apply heterogeneous GNNs for fraud detection from account-device graphs [54]; and Devign learns program semantics from code property graphs for vulnerability detection [55]. These studies highlight how focused subgraphs improve both accuracy and interpretability.

Subgraph definitions vary: some use fixed-radius k-hop neighborhoods, others employ attention or learned extractors. In this project, subgraphs are generated as company-centred snapshots within a fixed look-back window, ensuring only pre-announcement information is included.

Financial applications of this approach remain limited, with most GNNs still applied to static knowledge graphs or document-level graphs that neglect temporal variation. Recent advances such as Temporal Graph Networks (TGN) show that dynamically constructed subgraphs can better capture evolving influences [47].

Subgraph-based classification therefore provides a promising direction for modelling financial contexts. By focusing on local, semantically rich regions, it supports more accurate and interpretable GNN predictions. In this project, each subgraph snapshot embeds fact-level sentiment and event attributes alongside contemporaneous company metrics, serving as labelled instances for binary classification of earnings momentum.

2.5 Earnings Momentum and Surprise Modelling

Earnings momentum refers to the empirical tendency for stocks with strong earnings surprises to experience continued positive abnormal returns in the weeks and months following the announcement. This anomaly, most famously documented in the post-earnings-announcement drift (PEAD) literature, challenges the efficient market hypothesis and has inspired a large body of academic work (see Section 1.2.3).

The foundational studies by Bernard and Thomas [7] demonstrated that earnings surprises are not immediately and fully reflected in stock prices. Instead, prices continue to drift in the direction of the surprise over extended periods, a pattern that cannot be entirely explained by risk adjustments or information delays. This delayed response suggests that investors underreact to earnings announcements, causing gradual incorporation of information into prices (see Section 1.2.2).

Subsequent research has confirmed the robustness of this phenomenon. Jegadeesh and Livnat [56] showed that PEAD is stronger when firms exceed both earnings and revenue expectations, suggesting that the momentum signal strengthens when multiple financial indicators align. Similarly, Chan, Jegadeesh, and Lakonishok [9] found that the effect persists across different firm sizes and risk profiles, reinforcing its significance across market conditions.

Behavioural finance provides one explanation for this anomaly. Hong and Stein [10] proposed a model in which heterogeneous investor groups incorporate information at different speeds. Institutional investors may react quickly, while retail investors or analysts adjust more slowly. This staggered update leads to persistent price movement in the direction of the original earnings signal, the essence of earnings momentum.

This project builds upon these insights by predicting whether a firm will experience continued positive (or negative) abnormal returns in the weeks following an earnings announcement — the hallmark of earnings momentum described in the PEAD literature. Rather than forecasting the earnings surprise itself, the task is to identify earnings events where the initial surprise is sufficiently strong to generate a persistent momentum effect. To construct labels, we first measure the earnings surprise of firm i in quarter q as:

$$S_{i,q} = \frac{EPS_{i,q} - \hat{EPS}i,q}{\hat{EPS}i,q} \tag{2.1}$$

where $EPS_{i,q}$ is the reported earnings per share and $\hat{EPS}i,q$ is the consensus analyst forecast. A positive surprise occurs when $Si,q > 0$. We then compute post-announcement abnormal returns using the market-adjusted model:

$$AR_{i,t} = R_{i,t} - R_{m,t} \tag{2.2}$$

where $R_{i,t}$ is the return of firm i on day t and $R_{m,t}$ is the return of a market index (e.g., the S&P 500). The cumulative abnormal return (CAR) over the event window $[t_0, t_1]$ is then:

$$CAR_i(t) = \sum_{k=t_0}^{t} AR_{i,k}, \quad t \in [t_0, t_1] \tag{2.3}$$

The momentum effect is identified by estimating the gradient of CAR in the post-announcement period:

$$g_i = \frac{CAR_i(t_1) - CAR_i(t_0)}{t_1 - t_0} \tag{2.4}$$

Finally, a binary momentum label is assigned according to:

$$y_i = \begin{cases} 1 & \text{if } S_{i,q} > 0 \text{ and } g_i > \theta \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

where θ is a hyperparameter reflecting a meaningful threshold on the rate of drift. This ensures that only earnings events with both a positive surprise and a sustained upward gradient in post-announcement returns are labelled as momentum cases. These labels convert historical earnings announcements into supervised training data, where each instance is represented as a company-centred snapshot graph linking firm fundamentals with extracted financial events and sentiment signals.

3. Data Collection and Preparation

3.1 Company Selection and Price Coverage Filtering

The investable universe was defined from current S&P 500 constituents obtained from the public constituents table [57]. To ensure both breadth of financial disclosures and an abundance of related textual data, three of the most prominent sectors were selected: Information Technology, Consumer Discretionary, and Financials. These sectors represent a substantial share of overall market capitalisation, and companies within them are frequently the focus of both regulatory filings and extensive news coverage. Restricting the universe to these sectors therefore increases the likelihood of capturing rich interactions between firm fundamentals, narrative flow, and market response.

To ensure compatibility across all stages of the pipeline, the ticker universe was further restricted to symbols containing only alphanumeric characters. Certain S&P 500 constituents include special characters such as periods (e.g. `BRK.B`) or hyphens, which can introduce inconsistencies when interfacing with data vendors, parsing utilities, or database schemas. Restricting the universe to alphanumeric symbols eliminated these potential sources of error and ensured that all retained tickers could be processed uniformly across the SEC filings pipeline, market data downloads, and database storage.

To ensure sufficient historical coverage for consistent analysis across the study horizon, a price availability screen was applied. Firms were retained only if their trading histories contained the full number of required daily observations for the analysis window. Firms with incomplete histories were excluded from the metadata to avoid survivorship bias and truncation artefacts in later computations of abnormal returns around earnings announcements, which are central to evaluating earnings surprise impacts and PEAD.

3.2 Historical Price Collection and Benchmarking

Daily price series were retrieved programmatically for all companies in the defined asset universe using the `yfinance` Python package [58], which provides historical price data from Yahoo Finance. Auto-adjusted closing prices were collected to incorporate stock splits and dividends, ensuring comparability across time. A strict row-count requirement was enforced, whereby each retained ticker was required to exhibit the full complement of trading days over the configured study window. Firms with fewer observations were excluded to preserve consistency in event-window construction and baseline return estimation.

For each eligible ticker, new observations were appended to an SQLite database while preserving a composite primary key at the ticker–date level. This design ensures idempotent data updates and prevents duplication across re-runs. A short inter-request delay was introduced to reduce the likelihood of throttling by the data provider. In addition to the selected firms, the SPY exchange-traded fund (ETF) was included as a benchmark proxy for the U.S. equity market, used both for computing market-adjusted returns and for verifying calendar alignment.

The resulting dataset comprises uniformly adjusted, continuously sampled daily price series for all retained firms and the benchmark, with guaranteed alignment to the study window. This dataset provides the foundation for constructing event-centred return windows, computing abnormal returns, and evaluating the persistence of PEAD.

The process of company selection and historical price collection is illustrated in Figure A.1 in the Appendix. The diagram summarises the ideas conveyed in Sections 3.1 and 3.2, covering both the filtering of S&P 500 firms by sector and trading history, as well as the collection of adjusted daily prices through the `yfinance` package. This workflow ensures that only companies with abundant financial and news data are retained, while providing a consistent price history and an SPY benchmark for comparison.

3.3 SEC Filings Acquisition and Actual EPS Extraction

Accurate collection of Earnings Per Share (EPS) data is essential for quantifying earnings surprises and assessing post-earnings announcement drift (PEAD). Since the objective is to capture market reactions to quarterly announcements, EPS is required for all four fiscal quarters. Public firms release three 10-Q reports for Q1–Q3, each containing quarterly and year-to-date EPS, and a 10-K report for Q4 providing annual EPS. A full quarterly set is therefore constructed by combining the three 10-Q figures with the annual 10-K, inferring Q4 EPS as the difference between the annual value and the sum of the first three quarters. This feature engineering ensures consistent coverage across the fiscal cycle, particularly important given the influence of year-end results on PEAD effects.

The first stage of data acquisition involved building a complete index of relevant SEC filings. A Selenium-based scraper [59] was implemented to interact with the EDGAR portal [60], systematically retrieving 10-Q and 10-K filings for each ticker over the target date range. Metadata including ticker, form type, filing date, reporting date, and filing URL were extracted and stored in CSV format for subsequent EPS parsing.

Accession numbers were then derived from each URL. When combined with the ticker (e.g. `AAPL/0000320193-24-000081`), these identifiers allowed reproducible retrieval of parsed HTML filings via the `sec_downloader` package. The raw HTML documents were converted into semantic trees, enabling targeted searches for tabular EPS content. Regular expressions identified sections containing phrases such as "earnings per share" or "income per share," while excluding narrative references. This filtering ensured retention of only structured EPS tables.

A hybrid extraction approach was applied to these sections. Large language models (LLMs) were prompted with strict instructions to return values in the fixed format `basic_eps: value, diluted_eps: value`. Constrained formatting allowed automated validation and rejection of malformed responses. Multiple LLM backends were supported, with fallback mechanisms to handle rate limits. By aligning accession numbers, tickers, and extracted values, the final dataset provides a reproducible, machine-readable foundation for modelling earnings surprises and downstream PEAD dynamics. The pipeline is illustrated in Figure A.2 in the Appendix, which shows how raw filings are processed into structured quarterly EPS suitable for earnings surprise and momentum analysis.

3.4 Financial News Collection and Preparation

Financial news data was sourced from the FNSPID dataset, which provides a large collection of market-related articles with time-stamped metadata. The dataset was obtained from its official GitHub repository [61]. Since the raw files contained inconsistencies in formatting, the data was first cleaned to restore proper row structure and ensure article texts were usable. From there, only news items relating to companies in the selected S&P 500 sectors were retained, creating a filtered corpus of around 280,000 articles directly aligned with the earnings and price data described earlier. This provided a coherent news dataset that could be integrated into the knowledge graph construction process.

3.5 Consensus EPS Collection

Consensus EPS estimates reflect aggregated analyst expectations for a firm's quarterly performance and form the benchmark for calculating earnings surprises. These estimates are not freely available, as they require collecting and standardising forecasts from multiple market analysts. For this project, consensus data and realised surprises were purchased from Finnhub [62], which aggregates sell-side forecasts into harmonised quarterly and annual figures. This process ensures broad coverage and comparability across firms, making Finnhub a widely used source in financial research. Combined with the reported EPS values from 10-K and 10-Q filings in Section 3.3, these estimates enable consistent calculation of quarterly EPS surprises across the asset universe.

4. Implementation Knowledge Graph Assembly

4.1 Quantitative Market-Derived Features of Fact Nodes

To complement the textual and sentiment attributes of fact nodes, a set of quantitative market-derived features was incorporated. These metrics capture how firms were performing in the lead-up to an earnings announcement, ensuring that fact nodes represent not only event information but also contemporaneous market conditions. The features are grouped into four categories: momentum indicators, risk measures, market-adjusted performance, and technical signals.

Let $P_{i,t}$ denote the adjusted closing price of firm i on trading day t. The log return of the stock is defined as

$$r_{i,t} = \ln\left(\frac{P_{i,t}}{P_{i,t-1}}\right), \tag{4.1}$$

and the log return of the market index (e.g., S&P 500) is denoted $r_{m,t}$. All features are computed relative to the event date T, with different window lengths L corresponding to short- ($L = 20$), medium- ($L = 60$), or long-term ($L = 252$) horizons.

Momentum and return-based indicators

The cumulative return over a window of length L captures the total growth or decline of the stock:

$$CR_{i,L} = \sum_{t=T-L}^{T} r_{i,t}. \tag{4.2}$$

For example, $CR_{i,20}$ represents the cumulative log return of firm i over the 20 trading days leading up to the event.

A momentum gap is defined to measure acceleration or reversal in performance by comparing short- and medium-horizon returns:

$$MOM_i = CR_{i,10} - CR_{i,60}. \tag{4.3}$$

Here, $CR_{i,10}$ is the 10-day cumulative return, while $CR_{i,60}$ is the 60-day cumulative return. A positive value indicates short-term performance has outpaced medium-term performance.
Directional drift is further estimated by fitting a linear trend to log prices over the lookback window:

$$slope\_px_{i,L} = \arg\min_{\alpha,\beta} \sum_{t=T-L}^{T} \left(\ln P_{i,t} - (\alpha + \beta t)\right)^2, \tag{4.4}$$

where β measures the slope of the fitted line and reflects the average growth rate of log prices.
Finally, cumulative abnormal returns (CARs) measure firm-specific drift relative to the market benchmark:

$$CAR_{i,t} = \sum_{k=T-L}^{t} (r_{i,k} - r_{m,k}), \tag{4.5}$$

and the slope of the CAR curve provides the average rate of abnormal performance during the window:

$$slope\_CAR_{i,L} = \frac{CAR_{i,T} - CAR_{i,T-L}}{L}. \tag{4.6}$$

Risk and volatility measures

Volatility reflects the variability of returns. Standard volatility over L days is given by

$$\sigma_{i,L} = \sqrt{\frac{1}{L}\sum_{t=T-L}^{T}(r_{i,t} - \bar{r}_{i,L})^2}, \tag{4.7}$$

where $\bar{r}_{i,L}$ is the mean return in the window.
Downside volatility isolates only negative returns, penalising large losses more heavily:

$$\sigma^{-}_{i,L} = \sqrt{\frac{1}{L}\sum_{t=T-L}^{T}\min(r_{i,t}, 0)^2}. \tag{4.8}$$

Maximum drawdown quantifies the largest peak-to-trough loss experienced during the window:

$$MDD_{i,L} = \min_{t\in[T-L,T]}\left(\frac{P_{i,t}}{\max_{k\le t} P_{i,k}} - 1\right). \tag{4.9}$$

This provides a measure of tail risk that complements volatility.

Market-adjusted and residual performance

To isolate firm-specific behaviour from systematic factors, abnormal returns are defined as

$$abn_{i,t} = r_{i,t} - r_{m,t}, \tag{4.10}$$

with the cumulative abnormal return over the window given by

$$ABN\_SUM_{i,L} = \sum_{t=T-L}^{T} abn_{i,t}. \tag{4.11}$$

Regression against the market index separates systematic risk from idiosyncratic performance:

$$r_{i,t} = \alpha_i + \beta_i r_{m,t} + \epsilon_{i,t}, \tag{4.12}$$

where β_i is the market beta, α_i is the per-day abnormal return (alpha), and $\epsilon_{i,t}$ are residuals. The residual volatility

$$\hat{\sigma}_\epsilon = \sqrt{\frac{1}{L}\sum_{t=T-L}^{T}\epsilon_{i,t}^2} \tag{4.13}$$

measures idiosyncratic uncertainty unexplained by market movements. Market correlation is also included:

$$\rho_{i,L} = \mathrm{corr}(r_{i,t}, r_{m,t}), \quad t \in [T-L, T]. \tag{4.14}$$

Finally, the pre-event CAR measures abnormal performance in the 20 days directly preceding the event:

$$CAR^{pre}_{i,20} = \sum_{t=T-20}^{T-1} abn_{i,t}. \tag{4.15}$$

Technical and relative valuation indicators

Technical indicators capture well-established trading signals. The moving average gap measures divergence from recent trends:

$$MA\_gap_{i,L} = \frac{P_{i,T}}{\frac{1}{L}\sum_{t=T-L}^{T} P_{i,t}} - 1. \tag{4.16}$$

The Relative Strength Index (RSI), based on 14 days of average gains and losses, is defined as

$$RSI_{i,14} = 100 - \frac{100}{1 + \frac{\text{avg\_gain}_{14}}{\text{avg\_loss}_{14}}}, \tag{4.17}$$

where avg_gain_{14} is the average of positive price changes and avg_loss_{14} the average of negative changes.

The Moving Average Convergence Divergence (MACD) indicator compares short- and long-term exponential moving averages (EMAs):

$$MACD_i = EMA_{12}(P_{i,t}) - EMA_{26}(P_{i,t}), \tag{4.18}$$

with the 9-day EMA of the MACD providing a signal line:

$$Signal_i = EMA_9(MACD_i). \tag{4.19}$$

Finally, the relative distance to one-year extremes is computed as

$$pct\_to\_high = \frac{P_{i,T}}{\max_{t \in [T-252,T]} P_{i,t}} - 1, \tag{4.20}$$

$$pct\_from\_low = \frac{P_{i,T}}{\min_{t \in [T-252,T]} P_{i,t}} - 1. \tag{4.21}$$

These indicators summarise whether the stock is trading close to historical highs or lows at the event date.

By formalising these definitions and clearly linking each to financial intuition, fact nodes embed both qualitative event-derived sentiment and quantitative descriptors of firm performance. This dual representation allows the knowledge graph to capture the drivers of earnings momentum more effectively than text or prices alone.

4.2 Temporal Decay Coefficient

A key property of financial events is that their informational value diminishes as they become more distant in time. To capture this, each edge in the knowledge graph is assigned a temporal decay coefficient, denoted $\delta(\Delta t)$, where Δt represents the number of days between the event date and the earnings announcement date. The coefficient is a scalar $\delta(\Delta t) \in [0, 1]$ that downweights the contribution of older events when aggregating information in the graph. Formally, it satisfies the boundary conditions

$$\delta(0) = 1, \quad \delta(\Delta t) \to 0 \quad \text{as } \Delta t > \Delta t_{\max},$$

where $\Delta t_{\max}$ is the maximum look-back horizon (e.g., 90 days). Events with $\Delta t > \Delta t_{\max}$ are excluded from the snapshot graphs. A minimum retained value $\delta_{\min}$ is imposed at $\Delta t = \Delta t_{\max}$, ensuring the decay curves remain bounded before truncation.
In practice, the decay coefficient is defined piecewise: it follows the chosen functional form for $0 \leq \Delta t \leq \Delta t_{\max}$, and is set to zero thereafter:

$$\delta(\Delta t) = \begin{cases} f(\Delta t), & 0 \leq \Delta t \leq \Delta t_{\max}, \\ 0, & \Delta t > \Delta t_{\max}, \end{cases}$$

where $f(\Delta t)$ denotes one of several candidate decay functions.

Linear decay.

$$f_{\text{lin}}(\Delta t) = 1 - (1 - \delta_{\min}) \frac{\Delta t}{\Delta t_{\max}}$$

This form decreases linearly from 1 at $\Delta t = 0$ to $\delta_{\min}$ at $\Delta t = \Delta t_{\max}$, before being set to zero beyond the horizon.

Exponential decay.

$$f_{\exp}(\Delta t) = \exp(-\lambda \Delta t)\,,$$

where $\lambda = -\frac{\ln \delta_{\min}}{\Delta t_{\max}}$ is chosen so that $f_{\exp}(\Delta t_{\max}) = \delta_{\min}$. This produces a rapid early decay with a long tail.

Logarithmic decay.

$$f_{\log}(\Delta t) = 1 - \frac{(1-\delta_{\min})\ln(1+\Delta t)}{\ln(1+\Delta t_{\max})}$$

Here decay slows over time, reflecting cases where information remains useful for longer horizons.

Sigmoid decay.

$$f_{\text{sig}}(\Delta t) = \delta_{\min} + \frac{1-\delta_{\min}}{1+\exp\left(\alpha\left(\Delta t - \frac{\Delta t_{\max}}{2}\right)\right)},$$

where α controls the steepness of the drop-off. This form allows information to remain influential up to the midpoint of the horizon, then decline sharply.

Quadratic decay.

$$f_{\text{quad}}(\Delta t) = 1 - (1-\delta_{\min})\left(\frac{\Delta t}{\Delta t_{\max}}\right)^2$$

This emphasises more recent events while allowing a smoother decline than exponential.

These candidate functions provide flexible ways to model the diminishing value of financial events. A visualisation of these decays of a quarter (approx 90 days), towards a final value ($\delta_{\min}$) of 0.02 is shown in Figure A.3 in the Appendix.

4.3 Financial News Dataset Preparation

4.3.1 Exploratory Data Analysis

The cleaned news article dataset introduced in Section 3.4 provides a large-scale resource of approximately 280,000 articles. While this corpus forms a valuable foundation for building the financial knowledge graph, it also presents several challenges that limit its immediate usability.

No inter-company connectivity

A limitation of the dataset is that each article is tagged with only a single "primary ticker". This links every article exclusively to one company, even though in practice financial news frequently discusses multiple firms within the same piece. Because this happens so often, the dataset misses a large amount of cross-company context. As a result, any knowledge graph built directly from this data would be restricted to isolated sub-graphs centred on individual companies, failing to capture the inter-company dynamics that are essential for modelling how events propagate across the market.

This limitation was verified by applying regular expressions to the article text, using a mapping of key terms to company tickers, which allowed the creation of an additional "associated tickers" list for each article beyond the single primary label. Figure A.4 in the Appendix shows the distribution of article counts per ticker when only the primary ticker is considered. Nearly one third of the companies in the dataset have no articles in which they appear as the primary ticker, which means that for these firms it would not even be possible to construct a meaningful knowledge graph representation at all with this data format. On the other hand, if we look at Figure A.5, which reports the article counts after incorporating associated tickers, the coverage improves **dramatically**. Every asset now has significantly more connections, with each link representing a point where one company can be related to another through a shared article. Importantly, all companies now have at least one article

associated with them, and many of the firms that previously had zero primary ticker mentions are revealed to have hundreds of articles once these additional associations are taken into account.

Temporal Coverage Bias

Figure A.6 in the Appendix provides a scatter plot of all articles by date and primary ticker, sorted by volume. This plot highlights a structural flaw in the underlying dataset: a significant proportion of companies have no articles where they are the designated primary ticker. For these firms, it would not be possible to construct a knowledge graph at all, since no textual evidence is directly attributed to them. Even among companies with extensive coverage, such as Apple (AAPL), Microsoft (MSFT), Nvidia (NVDA), Tesla (TSLA), and Amazon (AMZN), most primary ticker articles are concentrated in only the last two to three years, introducing a temporal bias that skews coverage toward recent periods.

By contrast, Figure A.7 shows the same scatter plot but constructed from associated ticker mentions. When tickers linked to articles are taken into account, several issues are addressed simultaneously. First, it introduces inter-company connectivity, since multiple firms can now be linked through a single article. Second, each company benefits from a richer set of relevant information, including those with previously sparse article sets. Finally, incorporating associated tickers mitigates the temporal coverage bias observed in large-cap stocks, as older articles that reference these firms are surfaced alongside more recent ones.

Together, Figures A.6 and A.7 demonstrate that limiting analysis to primary tickers would leave large parts of the market unrepresented and temporally skewed, whereas including associated tickers provides a more connected, comprehensive, and temporally balanced foundation for downstream knowledge graph construction.

Granularity of Inter-Company Relations

While incorporating associated tickers significantly improves coverage and connectivity, it also introduces substantial noise if handled naively. Simply linking all companies mentioned in an article risks creating unjustified connections between firms that are not contextually related. For example, consider an article that states: *"Apple and Nvidia announce a new partnership, while Amazon reports strong quarterly earnings."* In this case, a valid link exists between Apple (AAPL) and Nvidia (NVDA), but Amazon (AMZN) should not be directly connected to either firm. If all three companies were linked indiscriminately, the resulting knowledge graph would contain spurious relationships that distort the structure of the network and undermine its utility for causal or temporal reasoning. This illustrates the need for finer-grained extraction methods that can distinguish between separate sub-stories within the same article and only create edges where a genuine inter-company relation is described.

This limitation motivated the introduction of an LLM-based approach to break down articles into finer-grained sub-stories, hereby known as facts, each capturing a specific piece of information and the companies directly involved. This is where the "LLM-Augmented" aspect of LLM-Augmented Knowledge Graph comes into play: rather than treating each article as a single indivisible unit, we process it into structured fragments of information that can be cleanly attributed to the relevant entities. By doing so, the bulky and often noisy article content is diluted into more granular and punchy bits of information, which substantially reduces noise in the KG while preserving contextually valid inter-company links. The result is a more faithful and precise representation of how events unfold across firms, laying a stronger foundation for downstream causal and temporal reasoning tasks. This step also marks a natural transition point in the pipeline, moving from raw dataset preparation toward true knowledge graph assembly.

4.3.2 LLM-Augmented Fact Extraction

Objective of Fact Extraction

With the case for fact-level representation established in Section 4.3.1, the objective at this stage is to define and operationalise what constitutes a "fact" within financial news, and to build a reliable mechanism for extracting such facts at scale. In this context, a fact is understood as the smallest

self-contained statement of financial relevance that can be attributed to one or more firms. Each fact must therefore capture the entities involved, the event being described, its temporal anchoring, and its sentiment.

The challenge is not only to identify these atomic units within often dense and discursive articles, but also to encode them in a structured form that is consistent enough for knowledge graph construction while remaining flexible to the variability of financial language. Large language models are employed here as the central tool for bridging unstructured text and structured representation, with prompt engineering and schema design ensuring that extracted facts are both machine-readable and semantically faithful to the original text.

To achieve this, each fact is represented using a deliberately compact schema comprising `date`, `tickers`, `raw_text`, `event_type`, and `sentiment`. This design strikes a balance between parsimony and expressiveness: the fields are sufficient to anchor facts temporally, attach them unambiguously to companies, describe the underlying event in natural language, and capture its evaluative tone, while avoiding unnecessary complexity that would introduce noise or increase the rate of schema violations. In this way, the schema provides a stable foundation for constructing a knowledge graph that is both interpretable and computationally tractable.

End-to-End Extraction Pipeline

The fact extraction pipeline transforms financial news articles into structured, sentiment-annotated fact records that can be directly incorporated into the knowledge graph. Its logic can be understood as a sequence of stages.

The process begins with a stream of cleaned articles, each associated with one or more firms through ticker symbols. Because articles frequently mention multiple companies or developments, a set of "focus tickers" is defined for each article, consisting of the primary firm and any associated firms detected in metadata. Facts are only extracted if they reference at least one of these focus tickers, ensuring that every record is attributable to the relevant entities.

For each article, a large language model is prompted with a structured instruction that guides the extraction of fact-level information. An example of the instruction given to the model is reproduced below:

```
You are an expert financial news analyst.
Your task is to extract, for each mentioned company, a structured, concise,
yet meaningful summary of the relevant story elements from the article.
Return your output as a JSON array where each element follows this schema:

{
  "date": "<YYYY-MM-DD>",
  "tickers": ["<TICKER_1>", ...],
  "raw_text": "<coherent description of the situation>",
  "event_type": "<string>",
  "sentiment": <float>
}

Rules:
- Each JSON object should group information relevant to a single ticker
  or a small set of closely linked tickers.
- DO NOT return any entry unless at least one of the focus tickers is mentioned.
- raw_text should be a coherent summary (1-3 sentences).
- event_type should be a concise snake_case descriptor such as
  "earnings_announcement", "partnership", "lawsuit", "product_launch".
- sentiment must be a float between -1 and 1.

Date: {date}
```

```
Article:
{article_text}

Respond with only a single JSON array or the format will be deemed invalid!
```

This prompt enforces a machine-readable schema while retaining flexibility in the assignment of `event_type` labels, which are not constrained to a fixed ontology. It also incorporates contextual constraints such as focus tickers, which restrict the extraction to firms of interest, and sentiment scores, which capture the tone of coverage for each fact.

Once the model returns a response, it is validated against the schema to ensure that every fact includes the required fields with the correct types. Only validated facts are retained. Each fact is annotated with the index of its source article to maintain provenance, and the collection of facts is written in line-delimited JSON format, where each line corresponds to a single fact object.

Through this pipeline, lengthy unstructured articles are reduced to compact, structured fact units. Each fact summarises a discrete financial development, ties it explicitly to the relevant firms, and quantifies its sentiment, providing a clean and auditable bridge between raw news text and graph-ready data.

Model Selection

Selecting an appropriate language model was a critical design decision in the development of the fact extraction pipeline. The task required a model capable of handling long financial articles, reliably producing schema-conformant JSON, and capturing subtle sentiment signals, while remaining efficient enough to scale across hundreds of thousands of documents.

A key requirement was that the model had to be deployed locally on available hardware (an Nvidia A100 GPU). Relying on paid API access to proprietary models such as GPT–5 would have incurred prohibitive costs, running into hundreds of pounds for repeated large-scale extraction, and would also have introduced external rate limits and reproducibility issues. This constraint meant the model needed to be both open-weight and sufficiently compact to run within local GPU memory. Models larger than 70 billion parameters were excluded, as they would not fit within available resources and would introduce unacceptable inference latency.

At the other end of the spectrum, smaller models, while faster, were found in preliminary trials to be less reliable at adhering to the structured output schema and often omitted relevant entities or produced malformed JSON. In addition, reasoning-augmented variants (so-called 'thinking' models) were rejected, as they introduced extraneous internal text into the output and significantly increased inference time without improving factual fidelity.

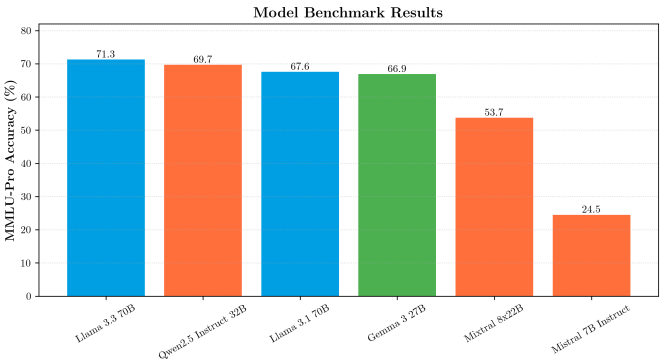


FIGURE 4.1: MMLU-Pro benchmark scores for candidate models considered in this project.

The candidate models considered were drawn from the highest-performing general-purpose reasoning models in the Ollama catalogue, which was selected for its ease of setup and built-in GPU optimisations. Within this set, Gemma 3 27B, Qwen 2.5 32B, Mistral 7B, and Llama 3.1 70B all satisfied the pre-requisites for local deployment and efficient large-scale inference. Their relative performance on the MMLU-Pro benchmark is shown in Figure 4.1. The results demonstrate that **Llama 3.3–70B achieved the highest score overall**, outperforming both other Llama variants and competing families such as Gemma and Qwen.

Balancing capability, schema adherence, and computational feasibility, the **Llama 3.3–70B** model was selected as the backbone for fact extraction. It represents a practical upper limit in model size, achieving state-of-the-art reasoning performance among open-weight models while remaining deployable on an Nvidia A100 via Ollama. This choice ensured that the pipeline could scale efficiently while maintaining reliable, structured, sentiment-aware fact extraction across the corpus.

4.3.3 Deduplication of Facts

An important preprocessing step in constructing reliable event-centric knowledge graphs is the removal of duplicate or near-duplicate facts. Without deduplication, the same piece of information may be counted multiple times, artificially inflating the importance of certain events and introducing bias into the graph structure.

The first level of deduplication is performed at the raw news article level prior to fact extraction. Articles that share identical titles (despite having different primary tickers) are treated as exact duplicates, since in most cases these represent syndicated press releases or repeated publications of the same story. In such cases, only one instance is retained.

A second layer of deduplication is applied at the fact level, after event extraction. Here, the similarity between fact representations is assessed by comparing their textual content. Each fact is represented by its *raw_text* (a short summary extracted from the source) and its associated *event_type*. These are embedded into a vector space, and the cosine similarity between two fact vectors x and y is computed as:

$$\cos(\theta) = \frac{x \cdot y}{\|x\| \, \|y\|} \tag{4.22}$$

where $x \cdot y$ is the dot product of the two vectors, and $\|x\|$ and $\|y\|$ are their Euclidean norms. This measure captures the orientation rather than the magnitude of vectors, and thus reflects the semantic similarity between two pieces of text. Facts whose embeddings exceed a pre-defined similarity threshold are flagged as near-duplicates, and only one representative is retained in the knowledge graph. This ensures that the graph captures unique events without over-representing repeated information, reducing noise and improving the robustness of downstream learning tasks.

4.4 Creating PEAD Classification Labels for Subgraphs

4.4.1 Quantifying the PEAD Time Window

To construct reliable graph labels, it was first necessary to determine the appropriate post–earnings announcement horizon over which abnormal returns should be measured. Following the literature on PEAD, labels are defined on the basis of cumulative abnormal returns (CAR) in a fixed event window after the disclosure of earnings results.

Earnings surprises were calculated by combining the realised EPS values collected in Section 3.3 with the consensus analyst forecasts described in Section 3.5. For each announcement, the surprise was taken as the difference between the realised EPS and the consensus EPS, divided by the absolute value of the consensus EPS. This produced a signed and normalised measure of earnings surprise, which was then used

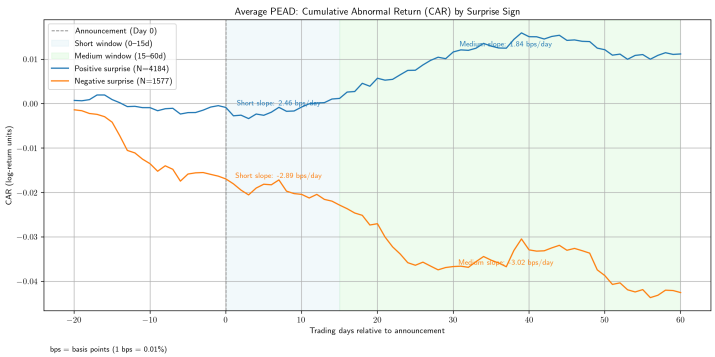


FIGURE 4.2: Average cumulative abnormal returns (CAR) following earnings announcements, split by positive and negative EPS surprises.

to classify announcements into positive and negative groups.

Using this classification, average CAR was computed across successive trading days following the announcement for both groups. This analysis served two purposes: it validated the presence of PEAD dynamics in the project dataset and guided the choice of an appropriate event window for label assignment.

The results, which are shown in Figure 4.2, confirmed the expected asymmetry. Firms with negative EPS surprises exhibited strongly negative CAR values from day 0 through day 60, showing a clear and persistent downward drift. By contrast, firms with positive EPS surprises displayed a more nuanced pattern: average CAR dipped below zero immediately after the announcement, began recovering around day 15, and tailing off by roughly day 40. This trajectory illustrates the gradual market underreaction that defines PEAD.

Based on these observations, the labelling horizon was fixed to the early drift period, capturing the divergence in returns between positive and negative surprise groups. Using this window ensures that the binary labels assigned to each earnings event accurately reflect the post–announcement momentum effect documented in both prior literature and in our own dataset.

4.4.2 Defining Positive PEAD Instances

We define a positive instance of post–earnings announcement drift (PEAD) at the firm–event level using cumulative abnormal returns (CAR) relative to a broad market benchmark. Let $t = 0$ denote the earnings announcement date. For each firm i we collect daily adjusted close prices for the stock and for the benchmark (SPY) and compute log returns:

$$r_t^{\text{stock}} = \ln\left(\frac{P_t^{\text{stock}}}{P_{t-1}^{\text{stock}}}\right), \qquad r_t^{\text{bench}} = \ln\left(\frac{P_t^{\text{bench}}}{P_{t-1}^{\text{bench}}}\right).$$

Abnormal returns are $a_t = r_t^{\text{stock}} - r_t^{\text{bench}}$ and cumulative abnormal returns are $\text{CAR}_t = \sum_{\tau \le t} a_\tau$.

The labelling criteria aim to isolate genuine post–earnings announcement drift (PEAD) rather than unrelated price movements. Three constraints are imposed. First, a positive EPS surprise is required; without it, strong post-announcement performance cannot be attributed to drift. Second, a medium-term slope threshold of 20 bps/day[1] ensures only firms with a clear upward drift above the test set average are classified as positive, while preserving class balance. Third, an overall positive slope from day 0 to day 40 excludes cases with an early sharp decline followed by rebound, as these do not reflect the gradual underreaction central to PEAD. Representative CAR trajectories for positive EPS cases are shown in Figures 4.3 and 4.4.

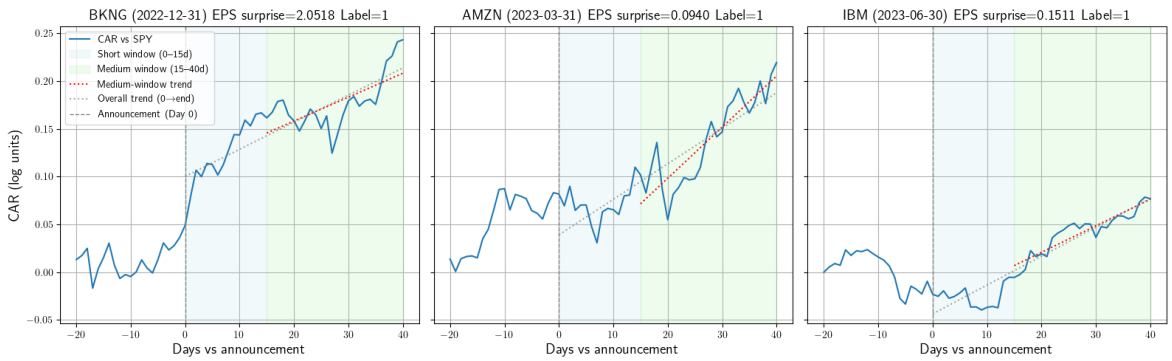


FIGURE 4.3: Examples of earnings events with positive EPS surprises classified as PEAD (label = 1).

[1] A basis point (bps) equals 0.01%, i.e. one hundredth of a percentage point. Thus, 20 bps corresponds to 0.20%.

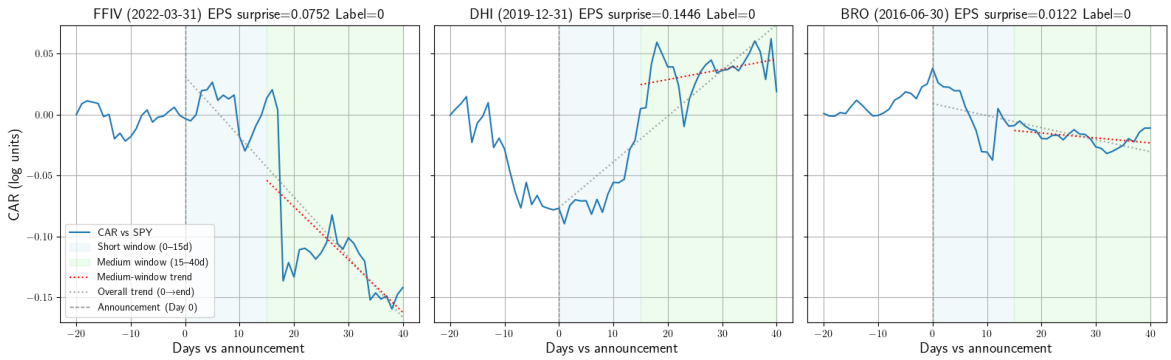


FIGURE 4.4: Examples of earnings events with positive EPS surprises classified as not PEAD (label = 0).

4.5 Final Subgraph Intermediate Representation

Each event is stored in an intermediate JSON representation that captures all information required for subgraph construction. The schema records the primary ticker, reporting date, and EPS surprise, along with a list of extracted fact nodes. Facts are drawn directly from the JSON schema introduced in Section 4.3.2, retrofitted with additional metadata such as the original fact index and temporal offset, before being placed into the event's `fact_list`. The representation also stores summary metadata including fact count and the final PEAD label.

For consistency, the fact list for each event covers the 90 days prior to the earnings announcement, approximately corresponding to one fiscal quarter. Since announcements across firms occur on aligned quarter-end dates, this design ensures that all subgraphs are constructed on an equal temporal footing. Restricting context to this pre-announcement window also helps to limit information leakage by preventing future events from influencing the representation.

Maintaining this text-based JSON format provides a clear and auditable checkpoint prior to encoding: it ensures that relevant facts are correctly linked to the focal firm, sentiment and event types are preserved, and the assigned label is visible alongside the supporting evidence. This intermediate layer both facilitates debugging and validates that the downstream graph will reflect the intended financial narratives.

4.5.1 Subgraph Exploratory Analysis

Of the 5,888 earnings announcement dates in the test set, 5,203 are associated with at least one fact, and 4,313 have ten or more associated facts. The distribution of fact counts is highly skewed: the mean fact list length is 66.93, while the median is 26. This indicates that while most events have a modest number of linked facts, a smaller subset of events accumulates substantially larger fact lists.

To ensure that each subgraph provides sufficient contextual information for classification, **a minimum threshold of 35 associated facts was imposed.** Subgraphs below this level were excluded on the basis that they do not contain enough evidence for a model to make an informed decision. Applying this threshold resulted in 2,456 subgraphs being retained for downstream experiments. A histogram showing the distribution of fact counts in these remaining graphs can be seen in Figure 4.5. For visual clarity, there are 129 subgraphs with more than 400 facts are not shown, these were removed as they are sparsely distributed, with the highest fact count being for a subgraph 2,848.

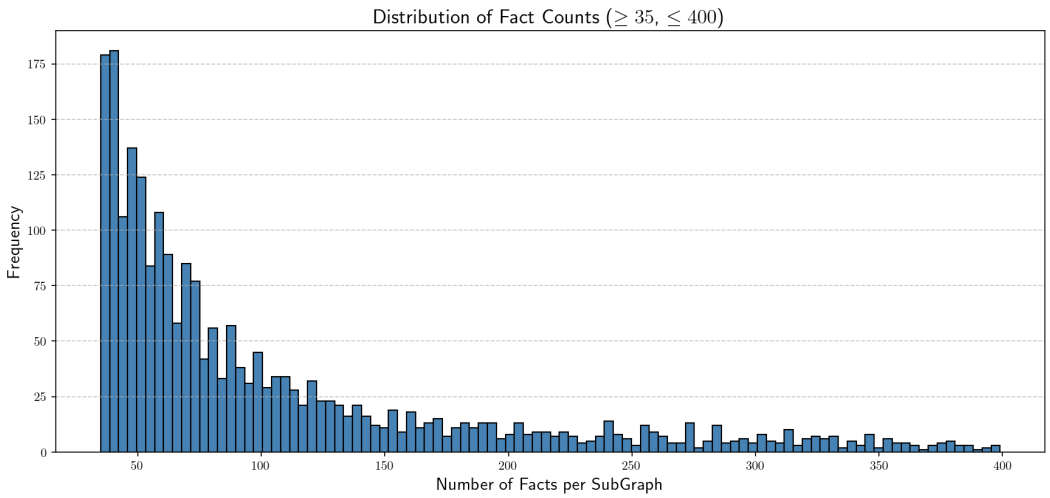


FIGURE 4.5: Distribution of fact counts across subgraphs with at least 35 associated facts.

4.5.2 Subgraph Label Distribution and Filtering

The distribution of labels across the retained subgraphs is imbalanced but informative. Among events with positive EPS surprises, 514 are labelled as PEAD (label $= 1$) and 1,317 as not PEAD (label $= 0$). In addition, 625 events with negative EPS surprises are always assigned label $= 0$ by construction. Since negative-EPS cases cannot correspond to positive drift, they are treated as noise and excluded from further experiments. After this filtering, a total of 1,831 subgraphs remain.

The filtered dataset therefore contains 514 positive instances and 1,317 negative instances, corresponding to a class imbalance ratio of approximately $1 : 2.56$. To mitigate the risk of the model being biased toward the majority class, this imbalance is incorporated directly into the training objective. Specifically, class weights are introduced into the binary cross-entropy (BCE) loss function, up-weighting the contribution of positive (label = 1) examples in proportion to the inverse of their frequency. This ensures that both classes exert comparable influence on the optimisation process despite their unequal prevalence.

4.5.3 Subgraph Visualisation

A custom visualisation tool was developed to render the intermediate JSON representations as node–edge diagrams, providing an intuitive view of the entities, facts, and labels within each subgraph before downstream modelling. This can be seen in Section A.4.

4.6 KGs/GNN Architecture Evolution

4.6.1 Theoretical Background on Knowledge Graph Representation

Knowledge Graph Basics

A knowledge graph (KG) is a structured representation of facts, where information is expressed in terms of entities and the relations between them. Formally, a KG can be defined as a directed multi-relational graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}) \tag{4.23}$$

where $\mathcal{V}$ is the set of nodes (entities or events), $\mathcal{E}$ is the set of edges, and $\mathcal{R}$ is the set of relation types. Each edge $e \in \mathcal{E}$ is a labeled connection (h, r, t) consisting of a head node $h \in \mathcal{V}$, a relation $r \in \mathcal{R}$, and a tail node $t \in \mathcal{V}$. Such triples (h, r, t) form the atomic unit of knowledge representation.

An equivalent algebraic view of a KG is given by adjacency matrices. For each relation $r \in \mathcal{R}$, an adjacency matrix

$$A_r \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|} \tag{4.24}$$

is defined, where $(A_r)_{ij} = 1$ if an edge of type r connects node i to node j, and 0 otherwise. In weighted or attributed graphs, these entries instead contain edge weights, such as sentiment scores, temporal decay factors, or learned time embeddings. Stacking all relation-specific adjacency matrices produces an adjacency tensor

$$\mathcal{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|\times|\mathcal{R}|} \tag{4.25}$$

which provides a compact way of encoding multi-relational structure and underpins many knowledge graph embedding and graph neural network methods.

In financial applications, nodes may correspond to firms, events, or documents, while edges denote relations such as "announces," "linked to," or "associated with." In this project, the KG departs from traditional entity-centric graphs by adopting an event-centric structure, where fact nodes encode textual and categorical attributes, and company nodes link these facts to quantitative financial metrics. Edges carry continuous weights derived from sentiment, temporal decay, and Time2Vec encodings, which enrich the adjacency representation with both semantic and temporal information.

Knowledge graphs may be either static or temporal. A static KG assumes that all relations are valid at once, while a temporal KG incorporates timestamps or decay functions that determine when relations hold. Temporal extensions are essential in finance, where the relevance of an event diminishes over time. This motivates the use of snapshot-based graphs, constructed at earnings announcement dates, that retain all relevant prior events within a defined time window.

Node and Edge Representations

The predictive performance of knowledge graph methods depends critically on how nodes and edges are represented in vector space. In this project, three main sources of information are encoded: fact node attributes, company node attributes, and edge attributes.

Fact nodes. Fact nodes correspond to atomic financial events extracted from textual data. Each fact node is assigned a continuous feature vector that combines two components:

- **Event type embedding:** categorical event classes (e.g., "earnings announcement," "analyst downgrade," "merger activity") are mapped to low-dimensional embeddings that capture their semantic role in the financial domain.
- **Text representation:** a dense embedding of the raw textual summary associated with the event, produced via a pre-trained language model. This captures contextual information beyond the discrete event label.

Together, these features allow the fact node to encode both structured and unstructured aspects of financial news.

Company nodes. Company nodes are instantiated at each earnings snapshot and contain quantitative descriptors of firm performance prior to the announcement. These features are derived from historical price and return series, covering cumulative returns, momentum gaps, volatility, drawdowns, abnormal returns, and technical indicators (see Section 4.1). Formally, each company node c is associated with a feature vector $x_c \in \mathbb{R}^d$ where each dimension corresponds to a financial metric relevant to earnings momentum.

Edges. Edges connect fact nodes to company nodes. They carry attributes that enrich the structural connections with semantic and temporal information:

- **Sentiment score:** a continuous measure of the positivity or negativity of the fact, derived from language model outputs.

- **Temporal decay coefficient:** a scalar value $\delta(\Delta t) \in [0, 1]$ that downweights the contribution of older events, where Δt is the number of days between the event date and the earnings announcement. The coefficient satisfies $\delta(0) = 1$ and $\delta(\Delta t) = 0$ for $\Delta t > \Delta t_{\max}$ (see Section 4.2).
- **Time2Vec embedding:** a learnable representation of the lag Δt, which captures both linear progression and periodic patterns in event timing [36].

The Time2Vec model is defined as

$$\text{Time2Vec}(\Delta t_{ij}) \;=\; \big[\, \omega_0 \Delta t_{ij} + \phi_0,\; \sin\big(\omega_1 \Delta t_{ij} + \phi_1\big),\; \ldots,\; \sin\big(\omega_K \Delta t_{ij} + \phi_K\big) \,\big], \tag{4.26}$$

where Δt is the time lag in days, ω_i and ϕ_i are learnable parameters, and $f(\Delta t)$ is the vector-valued embedding. The linear term ($i = 0$) captures monotonic progression over time, while the sinusoidal terms ($i > 0$) capture periodic or cyclical temporal patterns. Concatenating these outputs yields a flexible temporal representation that generalises the notion of positional encoding from transformers to continuous time.

In this project, edge attributes are represented by concatenating sentiment scores, temporal decay coefficients, and Time2Vec embeddings of the lag. This design allows the graph neural network to weight contributions not only by the importance of the event (sentiment) and its recency (decay) but also by learnable temporal dynamics captured through Time2Vec.

Heterogeneous semantics. By design, the graph is heterogeneous: it contains multiple node types (companies and facts) and edges with diverse attributes. Capturing these semantics is essential, as treating all nodes or edges uniformly would ignore crucial differences between financial entities and event-driven facts. Accordingly, the representation scheme ensures that both node-level features and edge-level modifiers contribute meaningfully to downstream learning tasks.

Message Passing in Graph Neural Networks

The central mechanism in graph neural networks (GNNs) is *message passing*, where the representation of a node is iteratively updated by aggregating information from its neighbors. This allows the model to capture both feature information and structural dependencies in the graph. Formally, if $h_i^{(l)} \in \mathbb{R}^d$ denotes the embedding of node i at layer l, then a generic message passing step can be expressed as

$$h_i^{(l+1)} \;=\; \sigma\left(U^{(l)}\left(h_i^{(l)},\; \sum_{j \in \mathcal{N}(i)} m^{(l)}\left(h_i^{(l)},\, h_j^{(l)},\, e_{ij} \right) \right) \right), \tag{4.27}$$

where $\mathcal{N}(i)$ is the neighborhood of i, $m^{(l)}$ is a learnable message function, $U^{(l)}$ is an update function, e_{ij} are edge features, and σ is a non-linearity.

From edge features to weights. Edges in the graph carry three attributes: a sentiment score $s_{ij} \in [-1, 1]$, a temporal decay coefficient $\delta(\Delta t_{ij}) \in [0, 1]$, and optionally a Time2Vec embedding of the raw lag Δt_{ij}. The decay coefficient is a scalar function of the time lag that downweights older events, with $\delta(0) = 1$ and $\delta(\Delta t_{ij}) = 0$ if $\Delta t_{ij} > \Delta t_{\max}$ (see Section 4.2). To derive a scalar edge weight $\tilde{a}_{ij}$, the features are concatenated into a vector

$$x_{ij} \;=\; \big[\, s_{ij},\; \delta(\Delta t_{ij}),\; \text{Time2Vec}(\Delta t_{ij}) \,\big], \tag{4.28}$$

which is passed through a small multi-layer perceptron (MLP) that outputs $\tilde{a}_{ij} \in (0, 1)$. This mechanism ensures that both the prevailing sentiment and recency of events, along with learnable temporal patterns, directly influence the flow of information through the graph.

Optional Time2Vec extension. Time2Vec maps the scalar lag Δt_{ij} into a vector of linear and periodic components, enabling the model to recognise temporal regularities as defined in Equation 4.26. In baseline models, only sentiment and decay are used, while extended models incorporate the Time2Vec features.

Propagation with weighted adjacency. Collecting the learned edge weights into a weighted adjacency matrix $\tilde{A}$, and applying symmetric degree normalisation, gives

$$\hat{A} \;=\; D^{-\frac{1}{2}}\,\tilde{A}\,D^{-\frac{1}{2}}, \qquad D_{ii} \;=\; \sum_{j}\tilde{a}_{ij}. \tag{4.29}$$

This normalised adjacency is then used in the standard Graph Convolutional Network (GCN) propagation rule, where node embeddings are updated as

$$H^{(l+1)} \;=\; \sigma\Big(\hat{A}\,H^{(l)}\,W^{(l)}\Big), \tag{4.30}$$

with $H^{(l)}$ denoting the matrix of node embeddings at layer l, $W^{(l)}$ a trainable weight matrix, and σ a non-linear activation. At the node level, this reduces to

$$h_i^{(l+1)} \;=\; \sigma\left(\sum_{j\in\mathcal{N}(i)}\hat{a}_{ij}\,W^{(l)}h_j^{(l)}\right). \tag{4.31}$$

Implementation note. In the PyTorch Geometric implementation, the edge encoder outputs $\tilde{a}_{ij}$ for each edge, which is passed as the `edge_weight` argument to `GraphConv`. This module applies the degree normalisation internally, ensuring consistency with the above formulation.

Graph Convolutions and Extensions

The propagation rules described above can be instantiated through different forms of graph convolution, each of which adapts the generic message-passing framework to the heterogeneity of the knowledge graph. In the present setting, two node types are present — *company* nodes and *fact* nodes — and a single relation type with bi-directional edges (*company–fact* and *fact–company*). This motivates the use of heterogeneous GNNs (HeteroGNNs), which treat messages differently depending on the source and destination node types, rather than assuming a homogeneous structure.

Graph Convolutional Networks. The standard Graph Convolutional Network (GCN) layer [63] operates on a normalised weighted adjacency matrix $\hat{A}$, updating node embeddings as

$$H^{(l+1)} = \sigma\Big(\hat{A}H^{(l)}W^{(l)}\Big), \tag{4.32}$$

where $H^{(l)}$ are the node embeddings at layer l, $W^{(l)}$ is a learnable transformation, and σ is a nonlinearity. Although originally designed for homogeneous graphs, this formulation provides the backbone for more advanced heterogeneous extensions.

Relational Graph Convolutions. In heterogeneous graphs, messages depend on the relation type. The Relational Graph Convolutional Network (R-GCN) [50] introduces relation-specific weight matrices:

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_r(i)}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)}\right), \tag{4.33}$$

where $\mathcal{R}$ is the set of relations, $\mathcal{N}_r(i)$ the neighbours of node i under relation r, $c_{i,r}$ a normalisation constant, and $W_r^{(l)}$ a relation-specific parameter matrix. In this framework, *company–fact* and *fact–company* edges can be treated with separate transformations, reflecting the potentially asymmetric importance of messages in each direction.

Attention Mechanisms. Relation-specific transformations capture heterogeneity, but not the variation in importance between individual neighbours. Attention-based approaches, such as the Graph

Attention Network (GAT) [64], weight each incoming message by a learned coefficient:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij} W h_j^{(l)}\right), \tag{4.34}$$

with coefficients α_{ij} computed via a softmax over neighbours. Heterogeneous extensions such as the Heterogeneous Graph Transformer (HGT) [51] combine relation- and type-specific projections with attention, allowing the model to adaptively emphasise more informative company–fact interactions. In financial settings, this aligns with the need to prioritise events with stronger sentiment or higher recency (i.e., larger decay coefficients).

Practical Adaptation. While transformer-style heterogeneous models are powerful, they are computationally intensive. In this project, a streamlined HeteroGNN is adopted, inspired by these ideas:

- bi-directional relation-specific aggregation for *company–fact* and *fact–company* edges,
- edge weights derived from sentiment, temporal decay coefficients $\delta(\Delta t)$, and optionally Time2Vec embeddings of Δt,
- simplified aggregation instead of full multi-head attention.

This design ensures that heterogeneity, sentiment, and temporal structure are all captured, while avoiding the computational overhead of full-scale relational or transformer-based architectures.

Readout and Graph-Level Representations

The downstream prediction task in this project is defined at the *graph level*, where each snapshot graph corresponds to a firm around an earnings announcement. After message passing, each node type has an embedding that encodes its local structural and feature context. To make a prediction, these node embeddings must be aggregated into a single vector representation of the snapshot graph. This step is known as *readout*.

In heterogeneous graphs, different node types carry distinct roles. Fact nodes represent individual financial events, while company nodes represent the firm context at the time of the announcement. Aggregating over all nodes without distinction risks obscuring these differences, so readout strategies are explicitly designed to separate and combine contributions from fact and company embeddings.

Formally, let $\mathcal{V}_{\text{fact}}$ denote the set of fact nodes and $\mathcal{V}_{\text{comp}}$ the set of company nodes in a graph. A generic global pooling operator $\rho(\cdot)$ maps a multiset of node embeddings to a single vector, e.g. mean pooling:

$$\rho(\{h_i : i \in \mathcal{V}\}) \;=\; \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} h_i.$$

Using this operator, four readout strategies are considered:

- **Fact readout:** aggregate only fact nodes,

  $$h_{\text{graph}} = \rho(\{h_i : i \in \mathcal{V}_{\text{fact}}\}),$$

  capturing event-driven signals while discarding explicit company features.

- **Company readout:** aggregate only the company node (restricted to the primary ticker at the announcement date),

  $$h_{\text{graph}} = h_c,$$

  where h_c is the embedding of the primary company node. This emphasises firm-specific descriptors while ignoring auxiliary events.

- **Concatenation:** pool both sets independently and concatenate,

$$h_{\text{graph}} = \left[\, \rho(\{h_i : i \in \mathcal{V}_{\text{fact}}\}),\ \rho(\{h_j : j \in \mathcal{V}_{\text{comp}}\}) \,\right],$$

  yielding a joint representation that combines fact-driven and company-driven signals.

These readout strategies reflect different inductive biases about how firm behaviour is influenced by event context. Fact-only readout assumes that market reactions are driven purely by external events, while company-only readout assumes internal firm characteristics dominate. Concatenation readout combines these perspectives.

Loss Functions and Class Imbalance

The prediction task in this project is binary classification, where the goal is to determine whether an earnings event will be followed by momentum. The natural choice of objective is the binary cross-entropy (BCE) loss. However, financial event data typically exhibits class imbalance: only a minority of earnings announcements lead to strong momentum. To account for this, a weighted version of BCE is employed, upweighting the contribution of positive (momentum) cases to the loss.

Formally, let $y_i \in \{0, 1\}$ denote the ground-truth label for sample i, and $\hat{y}_i$ the predicted logit. The weighted BCE loss is given by

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \Big(w\, y_i \log \sigma(\hat{y}_i) + (1 - y_i) \log(1 - \sigma(\hat{y}_i)) \Big), \tag{4.35}$$

where $\sigma(\cdot)$ is the sigmoid activation, and w is the positive class weight. The weight is chosen as

$$w = \frac{N_{\text{neg}}}{N_{\text{pos}}}, \tag{4.36}$$

with N_{pos} and N_{neg} denoting the number of positive and negative examples, respectively. This ensures that both classes contribute equally to the gradient signal, preventing the model from defaulting to the majority class.

The use of weighted BCE aligns with practices in financial prediction tasks, where extreme events are rare but of highest importance. In this project, the positive weight is computed directly from the training labels and passed into the optimisation routine.

Regularisation and Optimisation Strategies

Training graph neural networks on financial event data requires careful regularisation to avoid overfitting, given the relatively small number of earnings events compared to the dimensionality of the features. Several strategies are incorporated to improve generalisation.

Dropout. Dropout is applied at multiple levels of the architecture. Feature dropout randomly masks node embeddings during message passing, reducing reliance on any single feature dimension. Edge dropout is used as a structural form of regularisation: during training, a random subset of edges is removed, forcing the model to rely on multiple alternative paths for information propagation. This also serves as a data augmentation mechanism, introducing stochastic perturbations that encourage robustness to noisy or incomplete graph structure.

Optimisation. The models are optimised using the Adam optimiser [65], which combines adaptive learning rates with momentum through exponential moving averages of first and second moments of the gradient. Adam is widely adopted in GNN training due to its ability to handle sparse, noisy gradients and its reduced sensitivity to learning rate tuning. In some variants, AdamW (decoupled weight decay) can also be used to provide additional L_2 regularisation on the parameters, further improving stability.

Early stopping. Financial event prediction is prone to noise: small changes in news wording or timing can lead to volatile gradients. To prevent overfitting, early stopping is employed based on validation F1 score. The model is monitored over epochs, and training is halted if validation F1 does not improve for a fixed patience window. This criterion is preferable to accuracy or loss in imbalanced settings, as F1 balances precision and recall, directly reflecting the ability to identify momentum cases without overwhelming false positives.

4.6.2 Model 1: HeteroGNN (Baseline with Scalar Edge and Gated Readout)

The first architecture, denoted **heterognn**, serves as a baseline heterogeneous graph neural network. Its purpose is to test whether a simple scalar edge weighting derived from sentiment and temporal decay is sufficient to predict earnings momentum. The model operates over two node types: *fact* nodes and *company* nodes, connected by bidirectional edges.

Node representations. Fact nodes are encoded as the concatenation of two dense embeddings: one for the event type label and another for the raw textual description, both produced using the `all-mpnet-base-v2` sentence transformer. The resulting 1536-dimensional vector is linearly projected into a hidden dimension H, followed by ReLU activation and dropout. Company nodes contain 27-dimensional numerical descriptors derived from pre-event price and return series, which undergo the same projection, activation, and dropout to ensure compatibility with fact embeddings.

Edge representations. Edges carry two attributes: a sentiment score $s \in [-1, 1]$ and a temporal decay coefficient $\delta(\Delta t) \in [0, 1]$, where Δt is the number of days between the event and the earnings announcement. These attributes are combined through a simple gating function

$$g_{ij} = \sigma\big(w_1 s_{ij} + w_2 \delta(\Delta t_{ij}) + b\big)\,,$$

which outputs a scalar edge weight g_{ij} that modulates the strength of messages passed along the edge. This provides a lightweight mechanism to capture both prevailing sentiment and recency without additional temporal encodings.

Message passing. Message propagation is implemented with a `GraphConv` operator for each relation type (*fact*→*company* and *company*→*fact*). After each convolution, embeddings are passed through ReLU, Layer Normalisation, and dropout. This ensures stability and regularisation across all layers.

Readout. To obtain a fixed-size graph representation, several pooling strategies are considered:

- **Fact:** global mean pooling over all fact nodes.
- **Company:** mean pooling restricted to the designated primary company node (or all companies if no primary mask is available).
- **Concat:** concatenation of fact and company pooled representations.

Classification and training. The chosen readout embedding is passed through a linear layer with dropout and a sigmoid activation to predict the probability of positive earnings momentum. Weighted binary cross-entropy is used to correct for class imbalance. This design incorporates two complementary forms of gating: (i) *local* edge-level gating, which modulates the contribution of individual events based on sentiment and decay, and (ii) *global* readout-level gating, which balances reliance on textual event signals versus company fundamentals. Together with dropout applied throughout, this ensures robustness while keeping the architecture lightweight.

4.6.3 Model 2: Incorporating Temporal Encoding with Time2Vec

The second architecture, denoted **heterognn2**, extends the baseline **heterognn** by enriching the treatment of temporal information and introducing refinements aimed at stabilising training. In the baseline, each edge weight was determined by a simple logistic gating function over sentiment and a scalar decay coefficient $\delta(\Delta t)$, effectively collapsing temporal dynamics into a single recency score.

While this provided a coarse mechanism for down-weighting stale information, it could not capture richer temporal patterns or recurring structures in financial events.

In **heterognn2**, each edge is represented by the concatenation of three components: the sentiment score s_{ij}, the decay coefficient $\delta(\Delta t_{ij})$, and an optional Time2Vec embedding of the raw time lag Δt_{ij}. This yields the edge feature vector

$$x_{ij} = \left[s_{ij},\ \delta(\Delta t_{ij}),\ \text{Time2Vec}(\Delta t_{ij}) \right]. \tag{4.37}$$

These features are passed through a small multi-layer perceptron to produce the scalar edge weight:

$$w_{ij} = \sigma\big(\text{MLP}(x_{ij})\big)\,, \tag{4.38}$$

where σ denotes the sigmoid activation.

This design allows the model to jointly consider event sentiment, temporal decay, and richer periodic or seasonal patterns captured by Time2Vec. In this way, **heterognn2** conditions its message passing on both the diminishing importance of past events and on recurring temporal regularities, extending the baseline architecture without altering its overall structure.

4.6.4 Model 3: Temporal Control Baseline

The third architecture, denoted **heterognn3**, serves as a control variant to evaluate the impact of temporal encoding introduced in **heterognn2**. In contrast to the previous model, it omits all time-based features, relying solely on heterogeneous message passing between fact and company nodes.

Each edge retains only the sentiment component s_{ij}, yielding the simplified edge feature vector

$$x_{ij} = \left[s_{ij} \right]. \tag{4.39}$$

This is projected through a small multi-layer perceptron followed by a sigmoid to obtain the scalar edge weight:

$$w_{ij} = \sigma\big(\text{MLP}(x_{ij})\big)\,. \tag{4.40}$$

By removing both the decay coefficient and Time2Vec embeddings, **heterognn3** isolates the contribution of temporal information. Performance differences between this model and **heterognn2** therefore quantify the added predictive value of explicit temporal encoding, while keeping all other architectural components fixed.

4.6.5 Model 4: Attention-Based Message Passing

The fourth architecture, denoted **heterognn4**, extends **heterognn2** by replacing uniform aggregation with an attention mechanism. In earlier models, neighbour contributions were combined via weighted sums determined solely by edge features. While this incorporated sentiment, temporal decay, and optionally Time2Vec, all neighbours were still treated symmetrically once edge weights were computed. This assumption can be limiting in financial graphs, where certain facts may carry substantially greater predictive influence than others.

To address this, **heterognn4** incorporates Graph Attention Networks (GAT) [66], which learn attention coefficients to adaptively weight neighbour contributions during message passing. The update for a node i is given by

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j^{(l)} \right), \tag{4.41}$$

where $h_j^{(l)}$ is the embedding of neighbour j at layer l, W is a learnable transformation, and α_{ij} is the attention coefficient.

The attention coefficients are computed via a shared scoring function:

$$e_{ij} = \sigma\Big(a^\top [W h_i^{(l)} \,\|\, W h_j^{(l)} \,\|\, x_{ij}]\Big), \tag{4.42}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}(i)} \exp(e_{ik})}, \tag{4.43}$$

where a is a learnable attention vector, $\|$ denotes concatenation, and x_{ij} are the edge features (sentiment, decay coefficient, and optionally Time2Vec). This formulation allows the attention mechanism to condition not only on node features but also on edge attributes, directly aligning with the heterogeneous setting.

For expressiveness and stability, multi-head attention is employed:

$$h_i^{(l+1)} = \Big\|_{m=1}^{M} \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(m)} W^{(m)} h_j^{(l)}\right), \tag{4.44}$$

where M is the number of attention heads, each with its own parameters $W^{(m)}$ and coefficients $\alpha_{ij}^{(m)}$. The outputs of the heads are concatenated to form a richer representation of each node.

The readout stage remains consistent with previous models, supporting fact-only, company-only, or concatenated, pooling strategies. Dropout is applied both to the attention coefficients and to the transformed embeddings for regularisation.

The key innovation in **heterognn4** is therefore the introduction of neighbour-selective aggregation. Instead of relying solely on scalar edge weights (Section 4.6.3), the attention mechanism allows the network to dynamically prioritise the most influential events and company relationships. Temporal decay continues to provide a global discounting of older events, while attention introduces a relative weighting scheme among the remaining neighbours. Together, these mechanisms ensure that predictions are shaped both by the freshness of information and by its contextual relevance, providing a finer-grained treatment of event importance in earnings momentum forecasting.

4.6.6 Model 5: Edge-Aware GAT with Temperature, Entropy Sparsity, Top-k Pre-Gating, Buckets, and Jitter

The fifth model, denoted **heterognn5**, builds directly on the attention-based message passing of **heterognn4**, refining how edge information is weighted and filtered during training. The architectural backbone is unchanged: heterogeneous message passing with attention coefficients that incorporate sentiment, temporal decay, and optionally Time2Vec. Instead of altering the core propagation rule, **heterognn5** introduces several auxiliary mechanisms to sharpen attention, encourage sparsity, and improve robustness.

Attention Temperature Scaling. A temperature parameter $T > 0$ rescales edge logits before the softmax normalisation. For node i and neighbour j, let ε_{ij} denote the raw attention score:

$$\alpha_{ij} = \frac{\exp(\varepsilon_{ij}/T)}{\sum_{k\in\mathcal{N}(i)} \exp(\varepsilon_{ik}/T)}.$$

Smaller $T < 1$ sharpens the distribution, concentrating attention on a few neighbours, while larger $T > 1$ produces flatter weightings.

Entropy-Based Sparsity Regularisation. To discourage diffuse attention, the entropy of each attention distribution is penalised. For node i under head m:

$$H_i^{(m)} = -\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(m)} \log \alpha_{ij}^{(m)}.$$

The regulariser averages across all nodes and heads:

$$\mathcal{L}_{\text{entropy}} = \lambda_H \cdot \frac{1}{MH} \sum_{i,m} H_i^{(m)}.$$

This term is added to the main loss, encouraging sparsity so that only a small number of dominant neighbours remain influential.

Top-k Pre-Gating of Fact-to-Company Edges. To reduce neighbour crowding, each company node pre-selects the k most relevant fact edges using a heuristic score based on sentiment and decay:

$$\text{score}_{ij} = \delta(\Delta t_{ij}) \cdot \sigma(2s_{ij}),$$

where $\delta(\Delta t_{ij}) \in [0, 1]$ is the temporal decay coefficient (see Section 4.2), s_{ij} is the sentiment score, and σ is the logistic sigmoid. Only the top-k incoming edges with highest scores are retained.

Time Buckets. Alongside continuous temporal features (decay and Time2Vec), events are also mapped into discrete recency buckets. A set of thresholds $\{\tau_0, \tau_1, \ldots, \tau_B\}$ partitions delays:

$$b(\Delta t) = \min\{b \mid \Delta t \leq \tau_b\}.$$

Each bucket b has a trainable embedding u_b, appended to the edge feature vector. This gives the model a coarse-grained notion of temporal regimes (e.g., "very recent", "mid-term", "stale").

Train-Time Jitter (Edge Perturbation). To improve robustness and prevent overfitting to precise values, noise is injected into edge features during training:

$$s \leftarrow s + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_s^2),$$

$$\Delta t \leftarrow \Delta t \cdot (1 + \rho\eta), \quad \eta \sim \mathcal{N}(0, 1).$$

This perturbation affects both the decay coefficient $\delta(\Delta t)$ and the Time2Vec embeddings, forcing the model to generalise rather than rely on brittle feature values.

Together, these refinements make **heterognn5** an attention-based model with stronger control over neighbour weighting. Decay provides global down-weighting of stale events, attention selects the most relevant neighbours, and the auxiliary mechanisms (temperature, entropy, top-k, buckets, jitter) further constrain, sparsify, or diversify how attention is allocated. This enhances interpretability and stability while preserving the heterogeneous GNN foundation of earlier models.

4.7 Explainability

Explainability in machine learning refers to the ability to interpret and justify model outputs in human-understandable terms. For financial applications such as earnings momentum forecasting, explainability is particularly important because predictions must be accountable, auditable, and ultimately actionable by human analysts. Black-box predictions are insufficient; stakeholders need to understand which underlying factors contributed to a decision.

Knowledge graphs (KGs) provide a natural substrate for explainability, since their relational structure preserves the provenance of information in terms of entities, events, and links between them. Unlike unstructured models, KGs make explicit which facts are connected to a company's state and how information flows throughout. When combined with graph neural networks (GNNs), this structure enables attribution at the level of individual facts or event types. Attention-based architectures, such as GATv2 which is adopted in the final model iteration in Section 4.6.6, are particularly suited to this task because the learned attention coefficients can be directly interpreted as edge-level importance scores. In other words, the model itself provides a distribution over which facts most influenced the embedding of a company node, thereby giving an interpretable ranking of explanatory factors.

4.7.1 Event Clustering for Explainability

The original motivation behind the event_type field in the fact schema definition (first introduced in Section 4.3.2) was to support explainability by dividing facts into discrete, interpretable bins. In principle, this would enable model predictions to be traced back to specific categories of events, such as mergers, regulatory actions, or product announcements. However, implementing this approach directly would require the fact extraction pipeline to maintain a fixed taxonomy of all possible event categories at runtime. Given the open-ended nature of financial text and the difficulty of exhaustively defining such categories, this design was not feasible in practice.

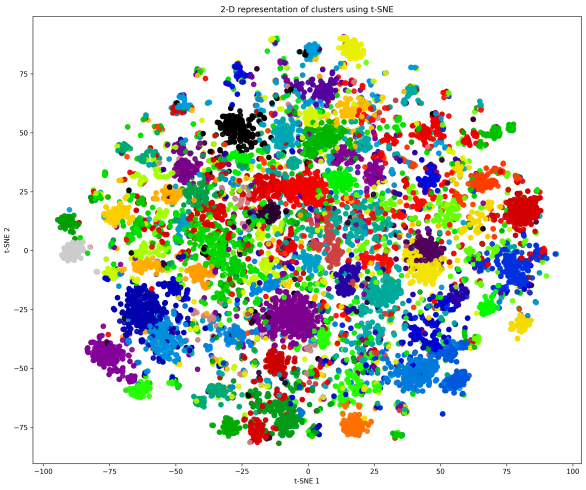


FIGURE 4.6: t-SNE projection of event-type embeddings, showing the 60 clustered groups used for explainability analysis.

To address this, the extraction process permits the large language model (LLM) unrestricted freedom in assigning event_type labels. These labels are subsequently embedded using the same encoder applied during KG construction, and the embeddings are clustered post hoc to form centroids that define discrete event bins. This approach provides a data-driven mechanism for grouping semantically similar events without constraining the pipeline to a predefined taxonomy, while also ensuring consistency with the latent space in which the GNN operates.

Figure 4.6 illustrates the distribution of clustered event_type embeddings in two dimensions using t-SNE. For the most part, it shows the formation of clear and distinct clusters, validating the efficacy of this methodolofy. Because the final model employs 60 clusters, the palette of distinguishable colours is limited; clusters that appear visually similar in colour but are separated in space correspond to distinct clusters. This representation highlights the semantic variety of extracted events and demonstrates the separation achieved by the clustering procedure.

Clustering thus enables explainability to operate at both the micro level (via attention weights on individual facts) and the macro level (via aggregated influence of clustered event categories). This allows systematic analysis of which types of events contribute most strongly to the model's predictions and supports a higher-level understanding of the patterns the model exploits.

4.7.2 Attention Weight Extraction

Attention weights in graph neural networks quantify the relative importance assigned to neighbouring nodes during message passing. In the proposed heterogeneous attention architecture, each fact connected to the primary company node contributes with a weight that is learned dynamically by the attention mechanism. These weights indicate how strongly individual facts influence the representation of the primary company, and by extension, the model's final prediction.

For explainability analysis, attention weights were captured from the final layer of Model 5 on the edges linking fact nodes to the primary company node. This ensures that only those connections contributing directly to the readout representation were considered. The extraction process collects the per-edge coefficients produced by the attention heads, which are then aggregated (by averaging across heads) to produce a scalar score for each fact. These scores form the basis for ranking and analysing which facts were most influential in the model's decision-making.

5. Results

5.1 Primitive Baselines

This section will only focus on samples where the EPS surprise is greater than zero. This is done because this is how the subgraph dataset was filtered in Section 4.5.2 (all graphs where EPS surprise < 0 were removed from the corpus). As a result, each baseline described below operates under the assumption that the earnings news itself is non-negative, and the task reduces to assessing whether additional information—most notably sentiment signals extracted from the surrounding textual facts—can help discriminate between cases where post-announcement drift aligns with the surprise and cases where it does not. This constraint simplifies the decision rules of the heuristics, since the sign of the EPS surprise need not be considered, and highlights the marginal contribution of sentiment in predicting drift direction within a uniformly positive-surprise setting.

5.1.1 EPS-always-positive Baseline

The simplest baseline assumes that a positive earnings surprise will always lead to drift aligned with the surprise. The classifier therefore applies a single deterministic rule: it predicts the positive class for every sample, regardless of the magnitude of the earnings surprise or any other features. This provides a trivial reference point against which more nuanced heuristics can be evaluated. Given the nature of the dataset, all predictions will be positive.

5.1.2 Sentiment Baseline

Given that the dataset has already been filtered to retain only events with positive earnings-per-share surprises (see Section 4.5.2), this baseline reduces the prediction task to a simple rule based solely on sentiment polarity. For each sample, the average sentiment score is computed across all associated facts. The classification rule is:

$$\text{predict 0 if } \overline{\text{sentiment}} \leq 0, \quad \text{otherwise predict 1.}$$

Here, label 1 denotes drift aligned with the (positive) earnings surprise, and label 0 denotes drift in the opposite direction. By construction, the EPS surprise no longer enters the decision rule, since its sign is always positive. This baseline therefore isolates the discriminative role of textual sentiment as the sole predictor of whether positive earnings news will translate into sustained positive drift in the post-announcement period.

5.1.3 Feedforward NN on Company Indicators

This baseline trains a feedforward neural network on the company-level indicators defined in Section 4.1, augmented with simple text-derived aggregates. For each event, the feature vector comprises: core financial indicators from the company node, together with lightweight sentiment summaries computed from associated facts (count of facts, total and average sentiment, counts of positive/negative/neutral facts), and basic EPS-derived scalars (raw, signed parts, absolute value). Inputs are standardised with a z-score scaler.

The classifier is a shallow multilayer perceptron with batch normalisation, ReLU activations, and dropout regularisation. Concretely, it stacks L hidden blocks (default $L = 3$), each block Linear $\rightarrow$ BatchNorm1d $\rightarrow$ ReLU $\rightarrow$ Dropout, with the hidden width halved after each block, followed by a single-logit output layer. The model is trained with BCEWithLogitsLoss, using a class-imbalance term `pos_weight` $= \frac{N_{\text{neg}}}{N_{\text{pos}}}$, optimised by Adam, and monitored with a validation split (stratified 80/20). Early stopping (patience 10) is triggered on the best validation loss. At inference, probabilities $\sigma(z)$ are thresholded at 0.5 to produce binary labels.

5.2 SOTA Baselines

5.2.1 Capturing Dynamics of PEAD with Genetic Algorithm-Optimised XGBoost [67]

Ye and Schuller (2020) investigate the predictability of PEAD by formulating the task as a binary classification problem. They predict the direction of the 30-day cumulative abnormal return (CAR) following an earnings announcement, using a supervised learning framework based on XGBoost with hyperparameters optimised by a genetic algorithm. Their feature set includes financial statement metrics, engineered changes, earnings surprise measures, technical indicators, and short interest data, applied to a large panel of Russell 1000 firms between 1997 and 2018. The model achieves up to 63% accuracy in predicting drift direction, with significant variation across sectors. Portfolio tests further demonstrate that stocks predicted to be in the top quantile generate positive abnormal returns, while those in the bottom quantile underperform, confirming the economic value of the predictive signal.

5.2.2 PEAD Prediction with Textual and Contextual Factors from Earnings Calls [68]

Chung and Tanaka-Ishii (2023) examine whether textual and contextual information from earnings calls can enhance the prediction of PEAD. They define the target as a binary classification task, identifying whether post-earnings returns over a 60-day horizon drift in the same direction as the earnings surprise. Their approach combines structured features (earnings metrics, fundamentals, technical indicators) with textual sentiment and readability scores, as well as contextual embeddings derived from sentence-BERT and ChatGPT-based summarisation. A blended model using CatBoost and logistic regression shows that contextual embeddings provide consistent incremental predictive power, improving both AUC (to approximately 0.55) and portfolio returns compared to structured features alone. The results highlight the usefulness of contextual language representations for capturing information not reflected in traditional financial variables.

5.3 Evaluation Metrics

To ensure a consistent and interpretable comparison across methods, all baseline heuristics and model architecture iterations are evaluated using five standard classification metrics: Accuracy, Recall, Precision, F_1, and AUC. These metrics capture complementary aspects of predictive performance:

- **Accuracy**: the proportion of all predictions that are correct, providing an overall measure of correctness.
- **Recall**: the proportion of true positives that are correctly identified, reflecting the model's sensitivity to positive cases.
- **Precision**: the proportion of predicted positives that are actually correct, reflecting the reliability of positive predictions.
- $\mathbf{F_1}$: the harmonic mean of precision and recall, balancing the trade-off between sensitivity and reliability.
- **AUC**: the area under the ROC (Receiver Operating Characteristic) curve, which plots the true positive rate against the false positive rate at varying thresholds. AUC measures the model's ability to discriminate between positive and negative cases across all possible classification cutoffs, with a value of 0.5 indicating random guessing and 1.0 indicating perfect separation.

The primitive baselines and the five successive iterations of the proposed model architecture are compared directly under this metric suite, providing a controlled view of incremental improvements. These metrics will be applied to each class individually, so a weighted average can be produced - addressing any issue in class imbalance. In addition, focused comparisons are carried out between the final model architecture and the state-of-the-art benchmarks from the literature.

5.4 Primitive Baseline Performance

Table 5.1 presents the performance of the three primitive baselines. As expected, the EPS-only heuristic performs poorly overall, with accuracy close to random and an AUC fixed at 0.5, since it always predicts the positive class. The sentiment-based heuristic introduces some discriminatory power, improving recall for the negative class, but still suffers from low precision on positives. The neural network baseline achieves the strongest results, with balanced recall across both classes and the highest weighted F_1 score, indicating that even simple feature-driven models can capture useful predictive signals beyond heuristic rules. These results serve as a baseline against which the proposed graph-based model architectures will be evaluated in the following sections.

		EPS	Sentiment	Neural Network
	Accuracy	0.281	0.367	0.527
	AUC	0.500	0.503	0.542
Class 1	Recall	1.000	0.811	0.575
	Precision	0.281	0.282	0.314
	F1	0.438	0.419	0.406
Class 0	Recall	0	0.195	0.508
	Precision	Undefined	0.726	0.754
	F1	Undefined	0.307	0.607
Weighted Average	Recall	0.281	0.368	0.527
	Precision	Undefined	0.601	0.630
	F1	Undefined	0.338	0.551

TABLE 5.1: Primitive baseline results

5.5 Performance of Model Architecture Iterations

During experimentation it was observed that results varied significantly between runs, depending on the random seed. To mitigate this and introduce stability, each architecture was trained 30 times with randomly initialised seeds on the same training and test splits. For evaluation, predictions from the 30 models were aggregated at the sample level, with the majority label assigned to each test case. Final performance metrics were then computed on these aggregated predictions, providing a more stable assessment of each architecture.

		Model 1	Model 2	Model 3	Model 4	Model 5
	Accuracy	0.779	0.789	0.772	0.795	0.805
	AUC	0.582	0.610	0.578	0.621	0.642
Class 1	Recall	0.191	0.255	0.191	0.277	0.319
	Precision	0.692	0.706	0.642	0.722	0.75
	F1	0.3	0.375	0.295	0.4	0.448
Class 0	Recall	0.972	0.965	0.964	0.965	0.965
	Precision	0.785	0.798	0.783	0.802	0.812
	F1	0.869	0.873	0.864	0.876	0.882
Weighted Average	Recall	0.753	0.766	0.748	0.772	0.784
	Precision	0.759	0.772	0.744	0.780	0.794
	F1	0.709	0.733	0.705	0.743	0.760

TABLE 5.2: Results of model iterations

Validation of Temporal Encoding A key finding from these experiments is that Model 3, which excluded any form of temporal encoding, performed the worst across almost all metrics. This underperformance highlights the importance of modelling temporal dynamics in financial knowledge graphs. Model 1, which relied on a single scalar to encode time, achieved reasonable performance but was consistently outperformed by Models 2, 4, and 5, which employed the richer Time2Vec encoding. These results demonstrate that temporal representations, and in particular the Time2Vec

approach, significantly enhance predictive performance and validate the hypothesis that earnings-related effects such as PEAD cannot be captured adequately without accounting for time.

High Precision and Low Recall for Class 1 A consistent pattern across all architectures is the imbalance between precision and recall for the positive class. Precision for Class 1 is relatively high, indicating that when the models do predict a PEAD event, they are usually correct. However, recall for Class 1 remains low, reflecting the difficulty of capturing all true positive cases. This behaviour can be attributed to the structured nature of the knowledge graph: strong predictive signals are only generated when there is sufficient evidence to perturb the graph representation, leading to many false negatives. While this limits the sensitivity of the models, in a financial setting it is more valuable to prioritise precision over recall. High precision ensures that the PEAD predictions generated are reliable and actionable, reducing the risk of costly false alarms in practical investment scenarios.

Robustness on the Negative Class Across all five architectures, performance on the negative class remained consistently strong. Recall for Class 0 was close to perfect in every case, and precision was also high, indicating that the models rarely produced false alarms. This stability demonstrates that the architectures are highly reliable at recognising non-PEAD cases, which is an important property in practice. In financial prediction, avoiding false positives is critical, as incorrectly signalling a drift event could lead to costly misinformed decisions. The consistent strength on the negative class therefore provides a solid foundation upon which improvements in positive-class prediction can be pursued.

Primitive Baseline Comparison When compared against the primitive baselines introduced earlier, all five architectures demonstrated a clear performance advantage across every metric. Even the weakest architectural variant (Model 3) outperformed the heuristic approaches, while the strongest design (Model 5) achieved substantial gains in AUC, weighted precision, and weighted recall. These results highlight the value of incorporating structured relational information and temporal encoding into the prediction task, confirming that graph-based architectures provide a decisive improvement over simple rule-based or feature-driven baselines.

Progressive Improvement Across Architectures A clear upward trend can be seen from Model 1 to Model 2, and continuing through Models 4 and 5, with each successive iteration introducing refinements that improved overall predictive capability. Model 4 introduced an attention mechanism over the knowledge graph, which significantly enhanced the model's ability to focus on informative substructures. Model 5 further extended this by incorporating a set of optimisations for attention, including an edge-aware graph attention network with temperature scaling, entropy-based sparsity, top-k pre-gating, bucket partitioning, and jitter regularisation. These refinements produced the strongest overall results. The progression is most clearly reflected in the steady increase of AUC, weighted precision, and weighted recall, which provide a fairer assessment under the 3:1 class imbalance. By contrast, raw accuracy appears superficially strong for all models, but is misleading in this context due to the heavy skew toward the negative class. The improvements across the weighted and ranking-based metrics indicate that the later architectures are not only more accurate in absolute terms, but also better calibrated to handle imbalanced financial prediction tasks. Model 5 achieved the highest overall performance across all evaluation metrics and is therefore used as the primary model for subsequent state-of-the-art comparisons and explainability studies.

5.6 Ensemble Diagnostics

This section presents a set of ensemble diagnostics designed to assess the stability and consistency of the training process across architectures. While the aggregate performance metrics provide an overall measure of predictive capability, they do not reveal how reliably individual models within the ensemble converge on the same decisions. To address this, diagnostics summarised in Tables 5.3 and 5.4 examine vote distributions, coverage of positive predictions, and the rate at which correct positives achieve majority support. These measures highlight differences in stability between architectures: earlier models often display greater variability, with fewer consistent majorities on positive samples, reflecting a limited ability to extract robust predictive signals. In contrast, the stronger architectures generate more stable and repeatable patterns of prediction, which is critical for ensuring

that the model's outputs are not overly sensitive to random initialisation or noise in the training process. These ideas will be explored in more detail in the remainder of this section.

	Total Sample Count	Number of Positives	Positives identified by any model	Coverage (%)	Majority positive vote count	Samples flagged as positive	Ensemble precision
Model 1	190	47	23	48.9	13	77	0.299
Model 2	190	47	23	48.9	17	47	0.489
Model 3	190	47	27	57.4	14	73	0.370
Model 4	190	47	19	40.4	18	34	0.559
Model 5	190	47	20	42.6	20	35	0.571

TABLE 5.3: Ensemble Diagnostic Results - Part 1

	Majority True Positive (TP)	Majority TP Rate (%)	Total Ensemble Score	Average Sample Score
Model 1	9	39.1	145.0	0.763
Model 2	12	52.2	148.1	0.779
Model 3	9	33.3	145.7	0.767
Model 4	13	68.4	149.0	0.784
Model 5	15	75.0	150.2	0.791

TABLE 5.4: Ensemble Diagnostic Results - Part 2

Ensemble Precision The third column of Table 5.3 reports the number of positive samples correctly identified by at least one model in the ensemble. Somewhat unexpectedly, Model 3 performs best on this metric, correctly identifying 27 out of 47 positives (57.4% coverage). By contrast, the attention-based architectures, which achieved the highest overall performance in Table 5.2, identified fewer positives, with 19 for Model 4 and 20 for Model 5. However, this difference is clarified when examining the sixth column, which shows the total number of samples flagged as positive by any model in the ensemble. Models 1 and 3 (those without Time2Vec) generated substantially more positive predictions overall, with 77 and 73 respectively, whereas the attention-based methods flagged only 34 and 35. This results in a much higher ensemble precision for Models 4 and 5, demonstrating that they are learning a more detailed and consistent representation of positive cases that is less dependent on parameter initialisation. By contrast, Models 1 and 3 display noisier behaviour, with a greater degree of bias introduced by random seed variability.

Majority True Positives Another important diagnostic is the rate at which correctly identified positive samples reach majority status within the ensemble. While the third column of Table 5.3 shows the total number of positives identified by any model, the first column of Table 5.4 reports how many of these were successfully promoted to majority agreement. Models 1 and 3 only managed to bring 9 positives to majority, from 23 and 27 identified respectively, corresponding to majority true positive rates of 39.1% and 33.3%. By contrast, the attention-based architectures show much stronger consistency: Model 4 promoted 13 of its 19 identified positives to majority (68.4%), while the optimised attention model (Model 5) achieved 15 out of 20 (75.0%). This demonstrates that the attention-based designs, and particularly Model 5, are learning representations of positive cases that are more robust to random parameter initialisation, leading to far greater stability in ensemble consensus.

This stability in training can further be visualised by tracking the average CAR of each model in the ensemble for each architecture. This can be seen in Figure 5.1 and it clearly shows that Models 1 and 3 display large variability, Models 2 and 4 show reduced but still noticeable spreads, and Model 5 exhibits a much narrower spread, indicating far more consistent training behaviour.

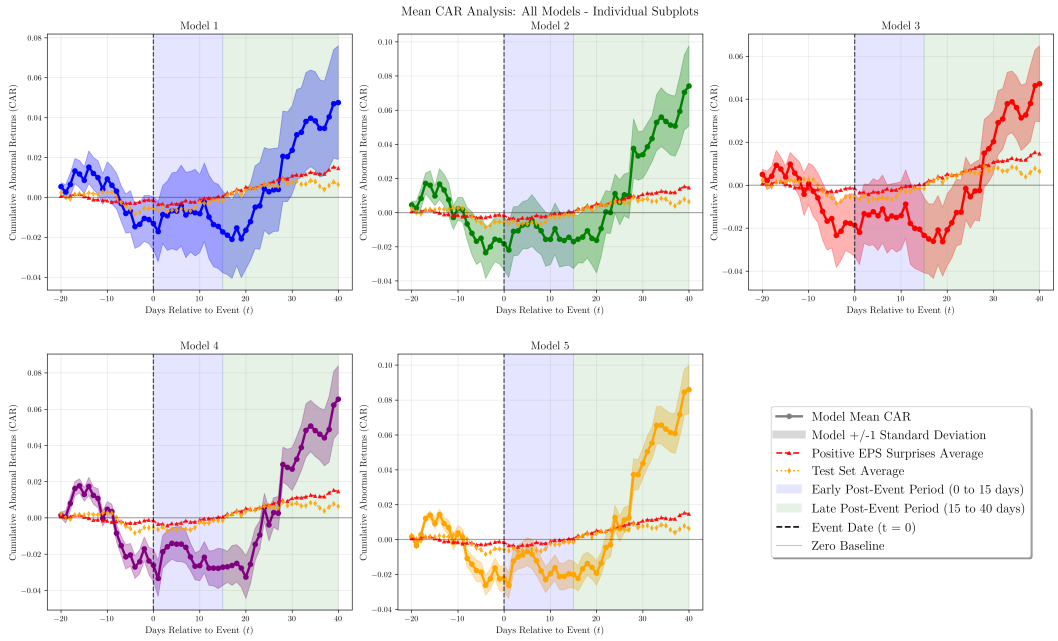


FIGURE 5.1: Average CAR trajectories around announcement date for positives identified by each architecture, with shaded areas showing ± 1 standard deviation.

Column 5 of Table 5.3 reports the total number of samples that each ensemble assigned a majority positive label, including both correct and incorrect cases. From this, the Class 1 precision values shown in Table 5.2 can be derived by dividing the majority true positives (Column 1 of Table 5.4) by this count. While not directly relevant to the ensemble stability analysis, this measure was included for completeness.

Overall Ensemble Performance The final set of diagnostics evaluates the overall performance of the ensembles using two related measures. For each sample i, the ensemble score is defined as the fraction of models that correctly predict its label:

$$s_i = \frac{\text{Number of models predicting correctly for sample } i}{\text{Total number of models in the ensemble}}.$$

The *Total Ensemble Score* is then the sum of these values across all N samples, while the *Average Sample Score* is given by the normalised mean:

These metrics capture both the accuracy and stability of the ensembles by accounting for the level of agreement across models. As shown in Table 5.4, performance generally increases across architectures, with Model 5 achieving the strongest results at an average sample score of 0.791. Interestingly, Model 3 (without temporal encoding) attains higher overall ensemble performance than Model 1, suggesting that the scalar temporal encoding used in Model 1 may not have aided training. More importantly, the steady improvements from Models 2, 4, and 5 demonstrate that the newer methods not only train more consistently, but also learn more accurate and robust representations of what a positive case of PEAD looks like.

5.6.1 Temporal Decay and Readout Methods

An ablation study on temporal decay functions found only minimal differences across variants, with exponential decay performing slightly better by more effectively down-weighting distant events. For readout strategies, Company pooling clearly outperformed Fact, and Concat alternatives, showing that firm-centred representations provide the most stable and discriminative signals. All reported results therefore use exponential decay with Company pooling, as this configuration yielded the most reliable performance.

5.7 SOTA Performance Comparison

5.7.1 Genetic Algorithm-Optimised XG- Boost

Ye and Schuller [67] approach the prediction of post-earnings announcement drift using genetic algorithm-optimised supervised learning applied to structured tabular features. Their study is evaluated primarily through directional accuracy and portfolio performance, with classification accuracy reported as the key diagnostic. On their dataset, covering over two decades of Russell 1000 earnings announcements, they report up to 63% accuracy in predicting drift direction, which corresponds approximately to an AUC of 0.63 under their balanced experimental setup.

The final architecture in this work achieves an AUC of 0.642, which is directly comparable to and slightly exceeds their reported classification benchmark. While their study benefits from substantial scale, made possible by relying on readily available structured tabular features, the present work operates under the more restrictive setting of qualitative event-based data. Such data is inherently scarcer and uneven across sectors, limiting sample breadth but enabling richer explainability through event structure and temporal relationships.

In addition to comparable predictive performance, the proposed method introduces clear advantages in terms of robustness and interpretability. Ensemble-based evaluation shows greater stability across random initialisations, and the temporal knowledge graph framework allows attention mechanisms to highlight the event clusters and time horizons most influential in classification. This contrasts with the tabular learning pipeline of Ye and Schuller, which emphasises scale and portfolio tradability but provides limited transparency into decision-making. Together, these results suggest that knowledge graph-based modelling provides a competitive and more interpretable alternative to feature-engineered tabular approaches for the PEAD prediction task.

5.7.2 Textual and Contextual Factors from Earnings Calls

Chung and Tanaka-Ishii [68] investigate the predictability of post-earnings announcement drift using textual and contextual information from earnings call transcripts, combined with fundamental and technical factors. Their study is primarily concerned with the economic value of predictions at the portfolio level, assessing abnormal returns generated by sorting firms into quantiles based on predicted drift intensity. Classification metrics are reported but serve mainly as a diagnostic of model performance rather than the central objective.

This framing contrasts with the present work, which is explicitly formulated as a binary classification problem: whether or not post-earnings announcement drift will occur. Despite the difference in emphasis, the classification results are directly comparable. Chung and Tanaka-Ishii report a best out-of-sample AUC of 0.549 when combining transcript embeddings with contextual features, whereas the final architecture in this work achieves an AUC of 0.642. This represents a substantial improvement in classification capacity, suggesting that the proposed knowledge graph-based approach learns more discriminative representations of PEAD than their transcript-driven logistic regression baseline.

It should also be noted that their approach benefits from scale, leveraging over a decade of earnings call transcripts across the S&P500. The dataset employed in this work is more constrained, since qualitative financial news is inherently scarcer and less uniformly distributed across sectors. Despite this limitation, the proposed architecture not only surpasses transcript-based methods in classification accuracy but also provides additional benefits in terms of temporal modelling, attention-based explainability, and ensemble stability. In this sense, the results highlight the value of temporal and event-structured knowledge graph methods as a complement to transcript-driven approaches.

5.7.3 Comparison to State-of-the-Art Approaches

Taken together, these comparisons show that the proposed knowledge graph-based approach achieves predictive performance that is at least comparable, and in some respects superior, to state-of-the-art baselines relying on structured tabular features [67] and earnings call transcripts [68].

While both prior studies benefit from scale and the relative abundance of their data sources, the present work operates under the more restrictive setting of qualitative financial news yet achieves higher classification AUC. Moreover, the integration of temporal encoding, attention mechanisms, and ensemble evaluation not only yields improved predictive performance but also introduces robustness and interpretability that extend beyond the capabilities of existing approaches.

5.8 Explainability Results

The analysis of attention weightings provides insight into which fact attributes were allocated the greatest importance during training. Two components are examined most closely: the `event_type`, which captures the nature of the underlying information, and the `date`, which encodes temporal context. Particular attention is also given to the large number of false negatives observed, which caused a low recall for Class 1 as seen in Table 5.2. Examining the distribution of attention in these cases helps to clarify why the models consistently struggled to identify certain positive samples of PEAD.

5.8.1 Event Type Attention Analysis

To investigate which categories of information were most influential in model decisions, attention distributions were analysed at the level of event clusters. For each model in the ensemble, samples were divided into those predicted as positive and those predicted as negative. Within each sample, the attention weight assigned to each `fact` $\rightarrow$ `primary_company` edge was extracted as an importance score for the corresponding fact. These scores were then aggregated across all positive and negative samples for all models in the ensemble. The aggregation was carried out by cluster identifier, producing average attention weightings, total fact counts, average sentiment scores, and descriptive text summaries based on the most common event types in each cluster. This procedure provides an interpretable view of which event categories consistently attracted model attention, and how their influence differed between positive and negative predictions.

Attention Patterns in Negative Classifications

Cluster ID	Total Fact Count	Average Attention Score	Average Sentiment Score	Cluster Summary
25	833	0.359	-0.507	consumer_spending_decrease economic_downturn market_decline
49	465	0.309	-0.440	geopolitical_risk export_restrictions tariff_impact
5	380	0.299	-0.146	demand_supply_mismatch production_challenges production_miss
21	409	0.263	-0.376	advertising_pause contract_loss customer_loss
55	1677	0.255	0.147	regulatory_action regulatory_announcement regulatory_challenge regulatory_approval

TABLE 5.5: Top 5 `event_type` clusters ranked by average fact attention score for samples with a negative predicted label

Table 5.5 presents the main attention weighting clusters associated with negative classifications. The clusters receiving the highest attention scores show strong thematic coherence, with most reflecting adverse economic, geopolitical, or operational conditions. Four out of the five clusters are associated with strongly negative sentiment, with average sentiment values ranging from –0.146 to –0.507.

These include macroeconomic downturn indicators such as consumer spending decreases and market decline (Cluster 25), geopolitical risks including export restrictions and tariff impacts (Cluster 49), production-side weaknesses such as supply–demand mismatches and production misses (Cluster 5), and operational losses including advertising pauses, contract terminations, and customer loss (Cluster 21). Each of these clusters is not only intuitively interpretable but also receives relatively high attention scores (0.263–0.359), indicating that the model systematically prioritises them when generating negative classifications.

It is also important to note the Total Fact Count column, which captures the number of times facts from each cluster appeared across all test set samples. These are consistently large values, ranging from several hundred to over 1,600, which reinforces the validity of the average attention scores. High fact counts reduce the risk that individual outliers dominate cluster statistics, ensuring that the derived attention weightings reflect genuine model behaviour rather than noise.

An exception arises with Cluster 55, which groups regulatory events such as regulatory actions, announcements, and approvals. Despite exhibiting a positive average sentiment score (+0.147), this cluster is frequently observed (1,677 facts in total) and still receives a strong average attention weighting (0.255). This suggests that the model does not rely solely on sentiment polarity when allocating attention, but instead recognises the structural and contextual importance of certain event types. Regulatory changes, while sometimes framed in neutral or positive language, often introduce uncertainty or impose constraints that align with negative market outcomes. Overall, the results in Table 5.5 show that the model assigns higher attention to clusters that represent credible sources of downside risk, with weightings that align with interpretable and meaningful event categories. The consistently large fact counts lend robustness to these findings, further supporting the conclusion that the model's attention mechanism captures substantive drivers of negative classifications.

Attention Patterns in Positive Classifications

Cluster ID	Total Fact Count	Average Attention Score	Average Sentiment Score	Cluster Summary
1	344	0.444	0.329	industry_comparison market_comparison comparison
45	89	0.441	0.578	new_launch promotion_launch
29	294	0.379	0.622	product_development product_launch product_adoption
22	174	0.352	0.370	executive_change executive_statement executive_movement
7	40	0.332	0.701	technology_transition technology_adoption

TABLE 5.6: Top 5 `event_type` clusters ranked by average fact attention score for samples with a positive predicted label

Table 5.6 reports the clusters that received the highest attention weightings in samples classified as positive. In contrast to the negatively weighted clusters described in Table 5.5, all five clusters here exhibit positive average sentiment scores, ranging from +0.329 to +0.701. This indicates that for positive predictions, the model tends to rely on sentiment polarity much more heavily. Thematically, these clusters reflect signals of growth, opportunity, and forward momentum. For example, Cluster 45 (new launches and promotions) and Cluster 29 (product development, product launches, and adoption) capture innovation and expansionary dynamics. Cluster 7 (technology transition and adoption) relates to technological competitiveness, while Cluster 22 (executive changes and strategic statements) highlights the importance of leadership shifts and corporate direction. Cluster 1, focused

on industry and market comparisons, represents relative positioning and benchmarking as another positive driver.

The attention scores for these clusters are notably high, ranging from 0.332 to 0.444, with Clusters 1 and 45 both exceeding 0.44. These values are generally higher than those seen for the top negative clusters, which suggests that when the model identifies a positive case, it relies on a smaller number of highly salient features rather than spreading weight across a broader evidence base. This pattern is consistent with the model's overall behaviour, where positive classifications are relatively rare and more concentrated in scope.

The Total Fact Count column shows that these clusters are supported by fewer examples than those associated with negative classifications, with values as low as 40 (Cluster 7) and the largest only reaching 344 (Cluster 1). This is not due to lack of relevance, but rather reflects the underlying class imbalance in the test set and the model's low recall for positive cases. In other words, because the model often fails to predict positive labels, fewer facts are aggregated into these clusters, reducing their representation. Despite this, the consistently high average attention scores suggest that when the model does predict a positive outcome, it does so based on a coherent and interpretable set of high-impact signals.

Overall, the results in Table 5.6 demonstrate that positive classifications are driven by sentiment-aligned, opportunity-oriented event types, with attention concentrated on relatively few but highly informative clusters. This highlights both the model's ability to isolate strong positive drivers of post-earnings drift and its difficulty in recognising such cases consistently across the dataset.

Comparative Analysis of Positive and Negative Attention

A comparison of Tables 5.5 and 5.6 highlights clear asymmetries between the clusters driving negative and positive classifications. For negative predictions, the clusters with the highest attention weightings are dominated by themes of risk and disruption, including macroeconomic decline (Cluster 25: consumer spending decrease, economic downturn, market decline), geopolitical risks (Cluster 49: export restrictions and tariff impacts), and production or operational challenges (Clusters 5 and 21). Four of the five clusters show strongly negative average sentiment values, although regulatory-related events (Cluster 55) receive substantial attention despite a slightly positive sentiment score. This indicates that the model is not purely sentiment-driven for negative classifications, but instead recognises structural signals of potential downside even in contexts where sentiment polarity is mixed. Importantly, the fact counts for these clusters are very large, ranging from several hundred to over 1,600, showing that negative classifications are supported by a wide and robust base of evidence across the test set.

By contrast, the clusters associated with positive classifications in Table 5.6 are fewer, smaller in fact count, and more tightly focused on themes of growth and opportunity. Examples include new launches and promotions (Cluster 45), product development and adoption (Cluster 29), technology transitions (Cluster 7), and executive changes (Cluster 22). Industry comparisons (Cluster 1) also feature prominently, with the model assigning higher attention where firms appear to outperform peers. All five clusters show positive average sentiment scores, with values ranging from +0.329 to +0.701, confirming that sentiment polarity plays a central role in driving positive classifications. Attention scores for these clusters are also higher than those for their negative counterparts, suggesting that when positive cases are identified, the model concentrates its focus on a smaller set of highly salient features rather than distributing weight broadly. However, the total fact counts for positive clusters are substantially lower, with some below 100, reflecting both the class imbalance in the dataset and the model's low recall for positive outcomes.

Taken together, these findings suggest that the model has developed two distinct regimes of reasoning. Negative classifications are broad, high-coverage, and sentiment-agnostic, reflecting diverse signals of risk. Positive classifications, in contrast, are rare, highly sentiment-driven, and focused on select indicators of growth or opportunity. This asymmetry explains both the model's relatively strong performance in identifying negative cases and its difficulty in consistently capturing positive cases of post-earnings announcement drift.

5.8.2 Temporal Attention and Date Influence

Figure 5.2 plots the aggregated attention weightings of all facts in the test set against their temporal distance from the associated earnings announcement. Each point represents the attention allocated to a fact–company edge across ensemble runs. The distribution appears noisy, with no obvious density gradient, reflecting the diverse contexts of facts. Even so, clear structural bands emerge: strong concentrations at 0.0 and 1.0, a dense band between 0.1–0.4, a weaker band near 0.5, and scattered points between 0.7–0.9.

Despite this variability, the fitted linear trend shows a negative correlation between attention and temporal distance (–0.248). On average, facts closer to the announcement are weighted more heavily, while distant facts are downweighted. Although the correlation is modest, the trend aligns with expectations that temporally proximate information is more relevant for modelling post-earnings announcement drift.

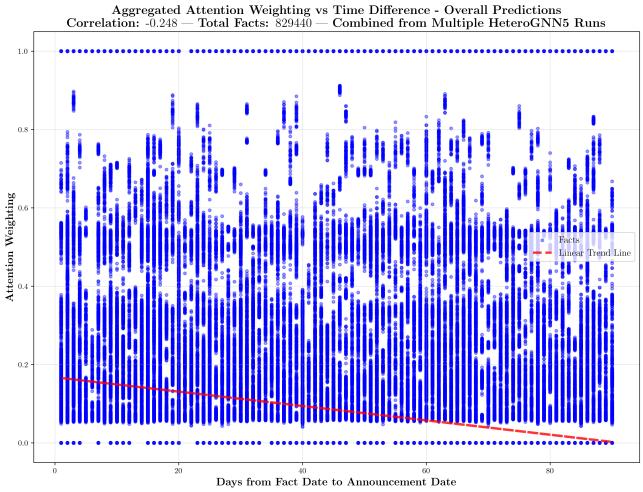


FIGURE 5.2: Scatter plot of aggregated attention weightings versus time difference to the announcement date.

While Figure 5.2 offers an overview of the relationship between attention weighting and temporal distance, it mainly illustrates the linear trend and is less effective for detailed analysis due to overlapping points. To address this, two complementary visualisations were constructed in Figure 5.3. First, a two-dimensional density function was generated by discretising both attention and time into bins, highlighting areas of concentrated weightings. Second, a box-and-whisker plot divided facts into time bins to show the mean, interquartile range, and overall range of attention values. Together, these provide a clearer view of how attention is distributed across temporal windows and support more precise analysis of temporal weighting behaviour.

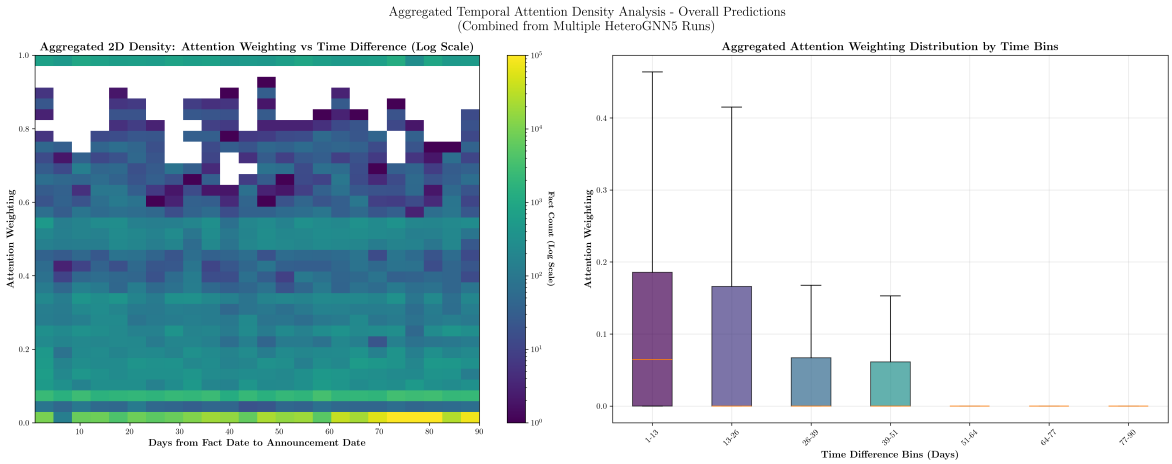


FIGURE 5.3: Density and distribution of attention weightings across temporal bins. Left: 2D density of attention versus time difference. Right: boxplot of attention distributions by time bin

Figure 5.3 shows the relationship between attention weighting and time difference via a 2D density plot. Several bands emerge. A strong line at 1.0 indicates a subset of facts consistently received maximal attention regardless of timing. Below this, a dense band in the 0.1–0.4 range captures most facts, with a slight gradient favouring more recent ones. The most notable pattern appears at 0.0, where a large number of facts 70–90 days before announcements are given zero attention—over an

order of magnitude more than in the 0–20 day window. Together, these results show that temporal proximity strongly shapes attention: recent facts are valued, while distant ones are often disregarded.

The same figure also presents box-and-whisker distributions across temporal bins. Attention is both stronger and more variable closer to announcements: the 1–13 and 13–26 day bins show wide, similar interquartile ranges (IQRs), while the 26–39 and 39–51 day bins narrow considerably. Beyond 50 days, the IQR collapses to zero, with medians and ranges converging at baseline. Although the density plot shows a persistent 1.0 band across all time spans, these appear as statistical outliers in the box-and-whisker convention. Overall, the results confirm that facts within the first month retain influence, while those older than 50 days are largely ignored.

5.8.3 Attention Patterns in False Negatives

Although the ensemble approach provides stability across random initialisations, a substantial fraction of positive cases were still unanimously misclassified as negative by the final architecture (Model 5). Specifically, 27 out of 47 positive samples (57.4%) were consistently assigned negative labels across all models in the ensemble. This large proportion of unanimous false negatives directly explains the very low Class 1 recall observed across all architectures. To investigate this behaviour, attention weightings for facts in these misclassified samples were aggregated and grouped by cluster. The objective is to identify which clusters consistently received high attention during these errors, and to evaluate whether these clusters align more closely with negative than positive predictive patterns. By comparing the relative rankings of these clusters in the negative and positive representation cluster rankings (NRCR and PRCR), it becomes possible to explain why genuinely positive cases were systematically suppressed into negative predictions.

Cluster ID	Total Fact Count	Average Attention Score	Average Sentiment Score	NRCR	PRCR
25	96	**0.735**	-0.333	**1**	29
49	64	**0.605**	-0.500	**2**	31
36	118	0.573	0.539	6	6
57	96	0.521	0.567	13	50
40	32	0.501	-0.525	40	54

TABLE 5.7: Top Clusters by Attention Score for Unanimous False Negative Predictions Across the Model 5 Ensemble

Table 5.7 highlights the top clusters by average attention score for positive samples unanimously misclassified as false negatives by the Model 5 ensemble. The most striking observation is the dominance of Clusters 25 and 49. These two clusters not only receive the highest attention weightings among all false negatives (0.735 and 0.605), but also occupy the top two positions in the negative representation cluster ranking (NRCR = 1 and 2). This aligns with the broader performance results, where the model achieves very high recall for Class 0 (negative cases). In practice, the model has become effective at spotting negative signals, consistently flagging facts from these clusters — such as consumer spending decreases, economic downturns, geopolitical risks, and tariff impacts — as indicative of a negative outcome.

Many of the misclassified positive samples contained numerous facts from these same clusters. Because the model assigned such high attention weightings to them, the ensemble consistently produced a negative classification even when the true label was positive. The remaining clusters (36, 57, and 40) also contribute, but their influence is weaker. Their rankings show that the model tends to interpret their structural patterns as negative even when sentiment is neutral or positive. Importantly, across all five clusters listed in Table 5.7, the NRCR is always greater than or equal to the PRCR, showing that these clusters are systematically ranked as more influential for negative predictions than for positive ones.

This analysis explains why recall for Class 1 remains low: positive cases containing events the model strongly associates with negative outcomes are systematically misclassified. The model is not ignoring the evidence but over-weighting signals it has learned to treat as reliably negative.

6. Discussion

6.1 Overview of Findings

The results presented in Chapter 5 highlight both the strengths and the limitations of the proposed methodology. The progression from primitive baselines to increasingly sophisticated graph neural network architectures demonstrated clear improvements in predictive capability, particularly with the integration of temporal encodings and attention mechanisms. These findings indicate that the framework is capable of capturing financially meaningful patterns, and the final architecture provided a compelling proof of concept for PEAD using LLM-augmented knowledge graphs.

At the same time, several constraints emerged. Most notably, the low recall for positive cases suggests that the feature set did not always provide sufficient granularity for the model to confidently identify momentum events, even though the precision of positive predictions was high. This points toward feature representation, rather than overfitting, as the primary bottleneck. Another important limitation relates to access to large-scale financial news corpora. While sufficient news data does exist, such as that available through Bloomberg terminals and other premium providers, it was prohibitively expensive for the scope of this project. As a result, the dataset used was less extensive and less balanced across sectors than would ideally have been the case.

Taken together, the work demonstrates both the promise and the challenges of graph-based approaches to PEAD forecasting. The following sections will consider the strengths of the proposed framework, its limitations, and the broader implications of the findings.

6.2 Strengths of the Proposed Approach

The results highlight several important strengths of the proposed methodology that together establish this work as a promising proof of concept for PEAD using temporal knowledge graphs.

Graph-based representation of financial context. A central strength lies in the use of knowledge graphs to represent both qualitative and quantitative information surrounding earnings announcements. This representation allowed the models to exploit structured relationships between companies, events, and time, in contrast to primitive baselines that relied on heuristics or isolated features. The improvement from the sentiment- and EPS-only baselines to the knowledge graph models demonstrates that the graph structure itself adds predictive power. Furthermore, the ensemble diagnostics showed that even early architectures could consistently identify non-trivial subsets of positive cases, confirming that the graph representation captures meaningful financial signals.

Temporal encoding improves predictive accuracy. Another key finding is the importance of temporal features. Results showed that Model 1, which used only a scalar temporal encoding, performed worse than Model 2, which introduced Time2Vec, while Model 3, which removed temporal information altogether, performed the worst of all architectures. This progression highlights that temporal structure is indispensable for the task, and that richer encodings such as Time2Vec allow the model to learn finer-grained decay patterns. The consistent improvements in AUC and weighted precision between Models 1, 2, 4, and 5 confirm that temporal encoding is not only useful but essential for capturing the dynamics of PEAD.

Attention mechanisms yield stability and refinement. The introduction of attention mechanisms in Model 4 provided a step change in performance. Compared to earlier models, Model 4 achieved stronger weighted precision and recall, reflecting its ability to prioritise the most relevant facts in the knowledge graph. Model 5 built on this with a series of optimisations—edge-aware GAT with temperature scaling, entropy sparsity, top-k pre-gating, bucketing, and jitter—that further improved performance and reduced training instability. Ensemble diagnostics reinforced this conclusion: while earlier models (particularly Models 1 and 3) displayed high variability across runs, Models 4 and 5 consistently converged on stable patterns and allocated fewer spurious positive predictions. This indicates that attention mechanisms not only boosted accuracy but also improved the robustness of the training process.

Explainability through attention weightings. The use of attention provided a built-in mechanism for explainability, a crucial requirement in financial domains where predictive outputs must be interpretable by human decision-makers. Analysis of attention weightings revealed that the models learned to distinguish between clusters of events with positive and negative associations, and that more recent events were systematically weighted more heavily than older ones. This transparency allows model outputs to be contextualised and provides insights into the causes of systematic errors, such as the consistent misclassification of positive samples dominated by strongly negative event clusters. The ability to interrogate the model in this way is a significant strength compared to traditional "black-box" methods.

Competitive performance despite data constraints. Perhaps most importantly, the final architecture achieved competitive performance relative to state-of-the-art baselines reported in the literature, despite operating under stricter data limitations. Published work has leveraged large-scale datasets such as earnings call transcripts (often proprietary and expensive to acquire), while this project relied on financial news and fact-level events extracted via LLMs. The high weighted precision and AUC achieved by Model 5 demonstrate that meaningful predictive power can still be extracted from smaller and less balanced datasets when organised in a graph-based structure. This suggests that the approach is not only effective but also adaptable to scenarios where access to premium data sources (e.g., Bloomberg terminals) is financially prohibitive.

In combination, these strengths establish the proposed framework as a robust and interpretable methodology for PEAD forecasting. While limitations remain, particularly with respect to recall, the results confirm that temporal knowledge graphs augmented with attention mechanisms provide a powerful foundation for capturing structured financial signals.

6.3 Limitations of the Proposed Approach

Despite the encouraging results, several limitations of the methodology must be acknowledged. These limitations help to contextualise the findings and also indicate clear directions for future work.

Low recall for positive cases. Across all architectures, the models consistently demonstrated low recall on the positive (class 1) cases. While precision was generally strong, the models frequently failed to identify true positive samples, resulting in a large number of false negatives. The explainability analysis suggests that this limitation arises from the model's tendency to heavily weight clusters of events that were historically associated with negative outcomes, even when they occurred in genuinely positive cases. This indicates that the feature set was not sufficiently rich to allow the model to differentiate subtle positive signals from dominant negative clusters.

Feature limitations in company nodes. The limited set of quantitative indicators incorporated into the company nodes constrained the expressiveness of the knowledge graphs. In practice, the dataset was restricted to signals that are primarily used in day trading, such as basic price movements, trading volume, and a handful of headline ratios. More sophisticated proprietary datasets could have provided far richer information. For example, liquidity measures (e.g., bid–ask spreads, order book depth), volatility-adjusted ratios, forward-looking analyst forecast revisions, or institutional ownership flows are available through platforms such as Bloomberg or Refinitiv but were inaccessible here due to cost. This shortcoming is likely a major factor behind the observed low recall, as the available features were not sufficient to capture the complex dynamics underlying drift. The limitation mirrors the issue faced with qualitative data, where large-scale financial news coverage existed but could not be accessed at industrial scale without proprietary sources.

Dependence on qualitative data availability. The qualitative side of the knowledge graph construction was constrained by data access. While sufficient financial news coverage exists in principle, large-scale access to professional datasets such as Bloomberg or Refinitiv was not possible within this project due to prohibitive costs. As a result, the dataset lacked full sectoral coverage, particularly in less frequently reported industries, which may have introduced bias. Although this limitation does not invalidate the results, it restricted the diversity of signals the models could exploit, particularly for cases outside the most heavily reported sectors.

Overfitting risk in attention-based models. Although Models 4 and 5 achieved the strongest performance overall, their reliance on highly expressive attention mechanisms carries a risk of overfitting, particularly given the limited size of the dataset. While the optimisations introduced in Model 5 (edge-aware GAT, entropy sparsity, pre-gating, and jitter) were designed to mitigate this, further work on regularisation and validation across multiple datasets would be necessary to confirm generalisability.

In summary, while the approach demonstrated promising predictive performance and interpretability, the limitations primarily stem from restricted features, constrained data access, and the challenge of class imbalance. These limitations also suggest clear avenues for future development, which are outlined in the following section.

6.4 Addressing the Research Questions

This section evaluates the extent to which the project has addressed its guiding research questions defined in Section 1.6.1. Each research question is revisited in turn, with results and analysis contextualised in relation to the objectives of the study.

RQ1: Can earnings momentum be effectively predicted using a snapshot-based knowledge graph constructed from unstructured qualitative financial text and quantitative financial data?

The results demonstrate that earnings momentum can indeed be predicted with reasonable success using knowledge graphs that combine unstructured qualitative text with quantitative indicators. All model architectures outperformed the primitive baselines by a substantial margin, showing that the structured representation of financial context provides predictive value. The best-performing architecture (Model 5) achieved an AUC of 0.642, alongside consistently higher weighted precision and recall compared to baseline heuristics. These results establish the feasibility of the knowledge graph approach as a proof of concept. At the same time, the persistently low recall highlights that the predictive signal is not yet fully captured, largely due to limitations in the feature set and the scarcity of large-scale proprietary data. Nonetheless, the models were able to capture meaningful patterns that indicate earnings momentum is at least partially predictable from the constructed graphs.

RQ2: Does incorporating sentiment and temporal decay improve the graph's ability to capture financially meaningful signals?

The comparative evaluation of architectures indicates that both sentiment and temporal encoding provide material improvements in predictive performance. Sentiment features, integrated through fact-level event nodes, enhanced the model's ability to differentiate between positive and negative drift scenarios by embedding sentiment directly into the graph. Temporal decay was evaluated across three configurations: scalar encoding (Model 1), Time2Vec (Models 2, 4, 5), and no temporal encoding (Model 3). Models employing Time2Vec consistently outperformed both scalar and absent temporal encodings, highlighting the importance of fine-grained temporal representation. The weakest overall performance was observed in Model 3, which omitted temporal signals entirely, further confirming their necessity. Together, these results show that sentiment and temporality are key drivers of financial signal capture within the graph framework.

RQ3: Can a graph neural network operating on snapshot subgraphs centred on earnings announcements learn predictive patterns from structured financial contexts?

The consistent improvement in predictive performance across model iterations confirms that GNNs are capable of extracting predictive patterns from subgraphs centred on earnings announcements. The progression from simple GCN layers to attention-based architectures shows that the models were increasingly able to exploit the structure of the knowledge graphs. Notably, the introduction of edge-aware attention with optimisations in Model 5 produced the strongest results, suggesting that more sophisticated GNN designs are well suited to financial subgraph prediction

tasks. Furthermore, ensemble diagnostics demonstrated that later architectures developed more stable and consistent representations of positive samples across different parameter initialisations. This evidences that GNNs are not only learning patterns from structured financial contexts but are doing so in a manner that is robust and generalisable.

RQ4: Does the use of fact-level events extracted by large language models provide a richer signal than traditional news sentiment scores?

The fact-level event extraction process enabled a more granular incorporation of qualitative information compared to traditional sentiment-based baselines. While simple sentiment aggregation could provide coarse predictive power, the knowledge graph construction allowed for nuanced event types, contextual sentiment, and temporal proximity to be integrated into the model. This richer representation was directly reflected in the comparative results, where all graph-based models significantly outperformed the sentiment baselines. Moreover, the explainability analysis revealed that the model consistently allocated higher attention weightings to particular event clusters (e.g., analyst commentary, corporate guidance), which would have been indistinguishable under traditional sentiment scoring. This shows that fact-level LLM extraction meaningfully enriched the qualitative signal available to the predictive models.

RQ5: To what extent can attribution help interpret the model's predictions, by identifying which event types, sentiment orientations, and recency windows were most influential?

The attribution analysis provided interpretable insights into the decision-making process of the models. Attention weighting analysis across clusters revealed clear distinctions between the types of events that were most associated with positive and negative classifications. For negative classifications, high attention was consistently given to clusters involving macroeconomic uncertainty, regulatory issues, or negative analyst commentary. For positive classifications, attention concentrated on clusters of corporate guidance and analyst upgrades. Temporal analysis showed a general decay in attention with distance from the earnings announcement, with more recent events carrying greater weight. These findings not only align with established financial theory but also help to explain the systematic bias toward false negatives: clusters strongly associated with negative outcomes received high weightings even in genuinely positive cases. Thus, attribution techniques proved valuable in interpreting the model's behaviour and in diagnosing weaknesses, providing a foundation for improving future iterations.

Summary

Taken together, the responses to the five research questions demonstrate that the project has successfully validated the central premise: knowledge graphs constructed from qualitative and quantitative data can capture meaningful financial signals that enable the prediction of earnings momentum. The iterative improvements across architectures showed the value of temporal encoding and advanced attention mechanisms, while attribution analysis confirmed that the models were learning patterns consistent with financial theory. At the same time, limitations in data availability and feature richness constrained recall, leaving scope for future improvements. Overall, the project establishes a strong proof of concept and offers a clear pathway for extending the methodology in subsequent research.

6.5 Implications for Financial Applications

The outcomes of this project carry several practical implications for financial applications, particularly in the domain of systematic trading, risk management, and corporate event analysis.

First, the performance profile of the models—characterised by consistently higher precision than recall—directly influences how such models would be deployed in practice. Precision in this context reflects the reliability of positive predictions of earnings momentum, meaning that when the model signals a likely drift, it is generally accurate. In financial terms, such a property is highly desirable, as false positives are costly; trading on a signal that proves to be incorrect can lead to direct financial

losses. By contrast, low recall, which reflects missed opportunities, is less damaging since investors can accept that not all events will be captured, provided that those flagged are of high quality. Thus, the models are positioned as tools for selective but dependable signal generation, complementing broader investment strategies rather than replacing them outright.

Second, the ensemble diagnostics demonstrated that more advanced architectures, particularly Model 5, not only achieved higher overall performance but also produced more consistent predictions across different parameter initialisations. This stability is critical for financial adoption. In practice, models must demonstrate robustness to random variations and training conditions; otherwise, their signals would not be trusted in high-stakes decision-making environments. The ability of attention-based models to repeatedly identify similar positive samples underlines their practical reliability and suggests that they could be used to construct portfolios with greater confidence in repeatable outcomes.

Third, interpretability emerges as a key enabler for integration into financial workflows. The explainability analysis, based on attention weightings, highlighted which types of events and temporal windows the models prioritised when forming predictions. For example, negative classifications often focused on clusters involving regulatory issues or macroeconomic uncertainty, while positive predictions were influenced by analyst commentary and corporate guidance. This aligns with established financial intuition and provides a degree of transparency necessary for compliance and stakeholder justification. Financial institutions increasingly require interpretable models to meet regulatory standards, and the approach used here provides a pathway to meet that requirement.

Fourth, the framework demonstrates the value of combining heterogeneous financial signals within a unified temporal knowledge graph. Traditional models often rely exclusively on either numerical indicators or aggregate sentiment scores. By contrast, the knowledge graph approach integrates earnings announcements, sentiment-rich news articles, analyst commentary, and basic quantitative indicators into a single structured representation. This integration enables the discovery of relationships and dependencies that would be invisible to models restricted to a single modality. From a financial applications standpoint, this represents an opportunity to improve event-driven investment strategies and to expand beyond earnings into other corporate events such as mergers, product launches, or regulatory decisions.

Finally, the study suggests that such models could form the basis of decision-support tools for analysts and fund managers. While they may not yet be ready for live trading given recall limitations, the models provide reliable 'flags' for high-confidence opportunities and can be used as an additional input into broader human- or algorithmic-driven strategies. Their explainability further allows analysts to interrogate predictions, ensuring that recommendations are grounded in identifiable financial contexts.

6.6 Future Work

There are several avenues for extending and strengthening this research in future work.

The most immediate priority is to address the limitations of feature representation, which were shown to have a significant impact on recall performance. The current models made reliable positive predictions when strong signals were present but struggled to detect subtler cases of earnings momentum. This suggests that the input features were not sufficiently detailed to capture the full range of financial drivers. Expanding the quantitative indicators embedded in the company nodes is therefore an important direction. At present, only a limited set of signals, many of which are commonly used in short-term trading, were available. In future work, richer financial metrics should be incorporated, including measures of institutional ownership, sector-adjusted valuation multiples, analyst consensus revisions, liquidity indicators, and volatility measures. These would provide a more comprehensive view of company fundamentals and market expectations, helping the model to distinguish positive cases that currently appear indistinguishable from negatives.

A second critical extension is to improve access to high-quality qualitative data. The scarcity of financial news available for training in this project was primarily a resource issue: while sufficient news coverage exists, large-scale access is prohibitively expensive. Proprietary datasets such as those available through Bloomberg or Refinitiv would allow for much deeper coverage of company-specific events, particularly for smaller firms and less-reported sectors that were underrepresented in the present dataset. With more complete qualitative data, the model would be better positioned to learn consistent signals across industries and avoid bias toward heavily reported firms.

A further important avenue is to enrich the structure of the knowledge graph by introducing additional node and edge types. The current design primarily captured companies, earnings events, and related textual facts, but financial systems are inherently more complex. Future work could integrate nodes representing institutional investors, supply chain partners, competitors, or regulatory bodies, each connected to firms through edges that denote relationships such as ownership, transactional links, sectoral competition, or oversight. Similarly, new edge types could represent subtler financial dynamics, including analyst forecast revisions, insider trading activity, cross-ownership ties, or co-movement of volatility across related firms. Incorporating this additional relational detail would allow the graph to encode a richer financial context, enabling the model to learn not only from direct company-level events but also from the broader ecosystem of interactions that shape earnings momentum.

Another promising direction is to explore extensions to the graph architecture itself. While the edge-aware attention mechanism with entropy sparsity and jitter demonstrated strong improvements, there remains scope to integrate more advanced temporal reasoning. Techniques such as temporal message passing with decaying memory, dynamic edge reweighting, or graph transformers could be tested to determine whether they capture temporal dependencies more effectively than current methods. Additionally, experimenting with multi-task objectives, where the model simultaneously predicts related financial outcomes such as volatility clustering or abnormal returns over different time windows, may improve generalisability and reduce overfitting to a narrow task.

Finally, improving the interpretability framework should be a priority. The attention-based analysis provided valuable insight into which event clusters and temporal windows were most influential, but future research could incorporate counterfactual reasoning, causal subgraph analysis, or perturbation-based methods to more rigorously quantify the role of specific events. These techniques would not only increase trust in the model's predictions but also help financial practitioners to understand the "why" behind the signals.

In summary, future work should concentrate on broadening the feature space, obtaining richer proprietary data, enriching the graph with additional node and edge types, testing advanced temporal architectures, and deepening the explainability framework. Together, these extensions would help improve recall performance while maintaining the high precision that makes the approach promising for financial applications.

6.7 Ethical Considerations

The application of machine learning models to financial prediction tasks raises several ethical issues that must be carefully considered. While the work presented here is framed as an academic proof of concept, the potential consequences of such approaches in real-world financial markets are significant, and so explicit reflection on ethics is required.

Data Accessibility and Fairness High-quality financial datasets, particularly comprehensive news feeds and earnings data, are prohibitively expensive. Platforms such as Bloomberg or Refinitiv provide the level of coverage needed to fully realise these models, but their costs make them accessible only to the largest institutions. As a result, smaller firms, individual investors, and academic researchers are systematically disadvantaged. This inequality risks concentrating predictive power in the hands of already-dominant market participants, which may widen existing divides in financial markets. The reliance in this project on more limited publicly available data sources reflects this imbalance and highlights how access to information is itself a major ethical issue.

Market Impact and Responsibility The potential market impacts and responsibilities of predictive systems must be recognised. If models of earnings momentum were to be widely adopted for speculative purposes, there is a risk they could amplify volatility. For example, if multiple firms act simultaneously on similar predicted signals, sudden inflows or outflows of capital could destabilise markets. Even if models are not perfectly accurate, the belief in their predictive power can influence behaviour, creating self-fulfilling or destabilising dynamics. It is therefore ethically important to frame such work carefully: the intent here is exploration and methodological development, not the immediate deployment of a trading system. Responsible use of such models should emphasise risk management, scenario analysis, and interpretability, rather than unchecked speculative application.

Transparency and Explainability Issues of transparency and explainability are central to the ethical deployment of financial AI. Models that act as "black boxes" undermine accountability, as users are unable to understand the reasoning behind predictions. This creates risks of over-reliance and blind trust in systems that may encode biases or errors. The analysis of attention weightings included in this project serves as an important countermeasure, providing at least partial interpretability of the model's decisions. This kind of transparency is not just scientifically useful but ethically necessary, allowing predictions to be scrutinised and questioned rather than accepted uncritically. Future work should build on this direction, exploring causal and counterfactual interpretability to make the decision-making of financial models more accountable.

Bias and Representativeness The issue of bias and representativeness must be acknowledged. Coverage of financial news is highly uneven across firms, with large-cap companies attracting far more reporting than small-cap firms. Similarly, structured financial data often prioritises large firms with greater market visibility. This introduces an inherent bias into the model's training, leading to predictions that are better calibrated for some companies than others. If deployed in practice, such biases could exacerbate unequal access to capital, as well-covered firms benefit from more accurate predictive signals while less visible firms are neglected. Making these biases explicit and transparent is itself an ethical responsibility, ensuring that readers understand the limitations of the model.

Research Ethics and Scope There are broader questions of research ethics and scope. This project is presented as an academic proof of concept, not as a system intended for immediate use in trading. It is important to make clear that the results here are exploratory, and any attempt to apply them directly to real financial markets would carry significant risk. Predictions generated by the model should not be taken as financial advice. Investing always involves uncertainty and the potential for loss, and any investor must perform their own due diligence before making decisions. Put simply, capital is always at risk, and responsibility ultimately lies with the individual making the investment. This principle of "do your own research" is fundamental and must be emphasised to avoid misinterpretation of the findings presented here.

6.8 Conclusion

This project set out to examine whether a structured, temporally evolving knowledge graph enriched with sentiment and causal information could improve the prediction of earnings momentum. The motivation arose from the limitations of existing approaches: traditional quantitative models overlook causal structure, while black-box machine learning methods often fail to provide transparency or interpretability. By contrast, an event-centric financial knowledge graph offered the potential to integrate heterogeneous information sources into a coherent, temporally grounded representation.

A key innovation of this work lay in the knowledge graph construction pipeline. Rather than treating news articles or filings as monolithic documents, they were collapsed into atomic fact-level events enriched with timestamps, sentiment, and event type attributes. These fact nodes were then linked to company nodes within dynamically constructed pre-announcement snapshots. This design ensured high information density while avoiding leakage of future information, as only data available up to the earnings release was included. Sentiment-weighted, time-decayed edges provided a principled way to model both informational tone and recency, while quantitative firm-level features were embedded alongside text-derived signals.

The graph-based learning framework built on these representations demonstrated that predictive accuracy could be enhanced by incorporating temporality, sentiment, and relational structure. Models trained on these subgraphs were able to capture patterns of post-earnings momentum that simpler baselines and flat feature sets could not. Importantly, the attribution module provided a measure of explainability, showing which event types, sentiment orientations, and temporal windows most strongly influenced predictions. This bridged the gap between accuracy and interpretability, aligning the methodology with the practical demands of financial applications.

The final model, which integrated Time2Vec encodings with an optimised edge-aware attention mechanism, highlighted both the strengths and current limitations of the approach. On the one hand, it achieved high precision in identifying positive momentum cases, meaning that when the model issued a positive prediction, it could be treated with a high degree of confidence. In a financial setting, this is a crucial property, as reducing false positives is often more valuable than maximising overall recall. The ensemble diagnostics further confirmed the reliability of this architecture: Model 5 not only produced the highest ensemble precision but also demonstrated the greatest stability in converging toward consistent representations of positive samples across random seeds. Tracking the average cumulative abnormal return (CAR) of predictions reinforced this stability, with Model 5 showing the lowest variance among all architectures.

On the other hand, recall for positive cases remained low, indicating that many genuine instances of momentum were not detected. The explainability analysis provided insight into this issue, showing that fact clusters heavily associated with negative outcomes often dominated attention weightings. When positive cases contained such events, the model tended to overrule weaker positive signals, resulting in systematic false negatives. This behaviour ultimately reflects the restricted feature space available in the dataset. The company nodes were populated with only a limited set of financial indicators typically accessible to retail traders, such as EPS surprises and basic accounting ratios, while the fact-level events were restricted to sentiment and simple event types derived from freely available news sources. In practice, more granular features—such as detailed analyst forecast revisions, capital flow data, or fine-grained causal dependencies between events—would likely increase the model's ability to detect subtle positive signals and thereby improve recall.

Taken together, these findings illustrate both the promise and the present boundaries of this approach. The results demonstrate that knowledge graph architectures can deliver predictive performance that is competitive with, and in some respects superior to, conventional baselines, while also offering interpretability. Moreover, when compared against recent state-of-the-art research methodologies tackling similar problems, the final architecture achieved stronger performance despite being trained on a more restricted dataset. The combination of high precision and low recall suggests that the framework is robust at confirming momentum when sufficient evidence is present, but struggles to uncover more ambiguous cases. This dual outcome underscores the potential of the methodology as a proof of concept: even with limited signals, the approach produced stable and interpretable predictions, and with richer data it is well-positioned to deliver significantly stronger performance in future work.

Bibliography

[1] Jason Fernando. "Earnings Per Share (EPS): What It Is and How to Calculate". In: (2025). URL: https://www.investopedia.com/terms/e/eps.asp.

[2] U.S. Securities and Exchange Commission. "Exchange Act Reporting and Registration". In: (2025). URL: https://www.sec.gov/resources-small-businesses/going-public/exchange-act-reporting-registration.

[3] Troy Segal. "Earnings Estimate: Meaning, Examples and Considerations". In: (2022). URL: https://www.investopedia.com/terms/e/earningsestimate.asp.

[4] Ben Mcclure. "Earnings Forecasts: A Primer". In: (2025). URL: https://www.investopedia.com/articles/stocks/06/earningsforecasts.asp.

[5] James Chen. "Consensus Estimate: Definition, How It Works, and Example". In: (2021). URL: https://www.investopedia.com/terms/c/consensusestimate.asp.

[6] James Chen. "Earnings Surprise: Overview, Examples, and Formulas". In: (2021). URL: https://www.investopedia.com/terms/e/earningssurprise.asp.

[7] Victor L. Bernard and Jacob K. Thomas. "Post-Earnings-Announcement Drift: Delayed Price Response or Risk Premium?" In: *Journal of Accounting Research* 27.Supplement (1989), pp. 1–36. DOI: 10.2307/2491062.

[8] Victor L. Bernard and Jacob K. Thomas. "Evidence that Stock Prices Do Not Fully Reflect the Implications of Current Earnings for Future Earnings". In: *Journal of Accounting and Economics* 13.4 (1990), pp. 305–340. DOI: 10.1016/0165-4101(90)90008-R.

[9] Louis K C Chan, Narasimhan Jegadeesh, and Josef Lakonishok. "Momentum Strategies". In: *The Journal of Finance* 51.5 (1996), pp. 1681–1713. ISSN: 0022-1082. DOI: 10.1111/j.1540-6261.1996.tb05222.x.

[10] Harrison Hong and Jeremy C. Stein. "A Unified Theory of Underreaction, Momentum Trading, and Overreaction in Asset Markets". In: *The Journal of Finance* 54.6 (1999), pp. 2143–2184. DOI: 10.1111/0022-1082.00184.

[11] Tobias J. Moskowitz, Yao Hua Ooi, and Lasse Pedersen. "Time Series Momentum". In: *Journal of Financial Economics* 104.2 (2012), pp. 228–250. DOI: 10.1016/j.jfineco.2011.11.003.

[12] Sheridan Titman and Narasimhan Jegadeesh. "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency". In: *Journal of Finance* 48 (Feb. 1993), pp. 65–91. DOI: 10.1111/j.1540-6261.1993.tb04702.x.

[13] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. "Deep learning for financial applications: A survey". In: *Applied Soft Computing* 93 (2020), p. 106384. DOI: 10.1016/j.asoc.2020.106384.

[14] Ming Jin et al. *A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection*. 2024. DOI: 10.48550/arXiv.2307.03759.

[15] Rakshit Trivedi et al. "Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. 2017, pp. 3462–3471. DOI: 10.48550/arXiv.1705.05742.

[16] Emanuele Rossi et al. "Temporal Graph Networks for Deep Learning on Dynamic Graphs". In: *arXiv preprint arXiv:2006.10637* (2020). DOI: 10.48550/arXiv.2006.10637.

[17] Didar Berk Araci. *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. arXiv preprint arXiv:1908.10063. 2019. DOI: 10.48550/arXiv.1908.10063.

[18] Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. "More than words: Quantifying language to measure firms' fundamentals". In: *The Journal of Finance* 63.3 (2008), pp. 1437–1467. DOI: 10.1111/j.1540-6261.2008.01362.x.

[19] Yi Yang, Ruslan Salakhutdinov, and William W Cohen. "FinBERT: A Pretrained Language Model for Financial Communications". In: *arXiv preprint arXiv:2006.08097* (2020). DOI: 10.48550/arXiv.2006.08097.

[20] Borui Cai et al. "Temporal Knowledge Graph Completion: A Survey". In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*. 2023, pp. 6545–6553. DOI: 10.24963/ijcai.2023/734.

[21] Weiqing Xu et al. "REST: Relational Event-driven Stock Trend Forecasting". In: *Proceedings of The Web Conference 2021 (WWW)*. 2021, pp. 1–10. DOI: 10.1145/3442381.3450032.

[22] Wenguang Wang et al. "Data Set and Evaluation of Automated Construction of Financial Knowledge Graph". In: *Data Intelligence* 3.3 (2021), pp. 418–443. DOI: 10.1162/dint_a_00108.

[23] Paul C. Tetlock. "Giving Content to Investor Sentiment: The Role of Media in the Stock Market". In: *The Journal of Finance* 62.3 (2007), pp. 1139–1168. DOI: 10.1111/j.1540-6261.2007.01232.x.

[24] Tim Loughran and Bill McDonald. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks". In: *The Journal of Finance* 66.1 (2011), pp. 35–65. DOI: 10.1111/j.1540-6261.2010.01625.x.

[25] Shimon Kogan et al. "Predicting Risk from Financial Reports with Regression". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*. 2009, pp. 272–280. URL: https://aclanthology.org/N09-1031.

[26] Xiao Ding et al. "Deep learning for event-driven stock prediction." In: *Ijcai*. Vol. 15. 2015, pp. 2327–2333.

[27] Xiaodong Ding et al. "Using Structured Events to Predict Stock Price Movement: An Empirical Investigation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1415–1425. DOI: 10.3115/v1/D14-1148.

[28] Ziniu Hu et al. "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 (WSDM)*. 2018, pp. 261–269. DOI: 10.1145/3159652.3159690.

[29] Johan Bollen, Huina Mao, and Xiaojun Zeng. "Twitter Mood Predicts the Stock Market". In: *Journal of Computational Science* 2.1 (2011), pp. 1–8. DOI: 10.1016/j.jocs.2010.12.007.

[30] Oren Etzioni et al. "Open Information Extraction from the Web". In: *Communications of the ACM*. Vol. 51. 12. 2008, pp. 68–74. DOI: 10.1145/1409360.1409378.

[31] David Nadeau and Satoshi Sekine. "A Survey of Named Entity Recognition and Classification". In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26. DOI: 10.1075/li.30.1.03nad.

[32] Yi Yang, Mark Christopher Siy UY, and Allen Huang. *FinBERT: A Pretrained Language Model for Financial Communications*. 2020. DOI: 10.48550/arXiv.2006.08097. arXiv: 2006.08097 [cs.CL].

[33] Hasan Hamad et al. "FIRE: A Dataset for Financial Relation Extraction". In: *Findings of the Association for Computational Linguistics: NAACL 2024*. 2024, pp. 3628–3642. DOI: 10.18653/v1/2024.findings-naacl.230.

[34] Rakshit Trivedi et al. "DyRep: Learning Representations over Dynamic Graphs". In: *International Conference on Learning Representations (ICLR)*. 2019. URL: https://openreview.net/forum?id=HyePrhR5KX.

[35] Da Xu et al. "Inductive Representation Learning on Temporal Graphs". In: *International Conference on Learning Representations (ICLR)*. 2020. URL: https://openreview.net/forum?id=rJeW1yHYwH.

[36] Seyed Mehran Kazemi et al. "Time2Vec: Learning a Vector Representation of Time". In: *arXiv preprint arXiv:1907.05321* (2019). DOI: 10.48550/arXiv.1907.05321. URL: https://arxiv.org/abs/1907.05321.

[37] Natthawut Kertkeidkachorn et al. "FinKG: A Core Financial Knowledge Graph for Financial Analysis". In: Feb. 2023, pp. 90–93. DOI: 10.1109/ICSC56153.2023.00020.

[38] Xiaohui Victor Li and Francesco Sanna Passino. "FinDKG: Dynamic Knowledge Graphs with Large Language Models for Detecting Global Trends in Financial Markets". In: *Proceedings of the 5th ACM International Conference on AI in Finance*. ACM, Nov. 2024, pp. 573–581. DOI: 10.1145/3677052.3698603. URL: http://dx.doi.org/10.1145/3677052.3698603.

[39] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901. DOI: 10.48550/arXiv.2005.14165.

[40] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *arXiv preprint arXiv:2203.02155* (2022). DOI: 10.48550/arXiv.2203.02155.

[41] Zexuan Ji et al. "A Survey of Hallucination in Natural Language Generation". In: *ACM Computing Surveys* (2023). DOI: 10.1145/3571730.

[42] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of NAACL-HLT*. 2019, pp. 4171–4186. DOI: 10.48550/arXiv.1810.04805.

[43] Yuan Yao et al. "DocRED: A Large-Scale Document-Level Relation Extraction Dataset". In: *Proceedings of ACL*. 2019. DOI: 10.18653/v1/P19-1074.

[44] Sheng Wu et al. "BloombergGPT: A Large Language Model for Finance". In: *arXiv preprint arXiv:2303.17564* (2023). DOI: 10.48550/arXiv.2303.17564.

[45] Pengfei Liu et al. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *arXiv preprint arXiv:2107.13586* (2021). DOI: 10.48550/arXiv.2107.13586.

[46] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. *FinGPT: Open-Source Financial Large Language Models*. 2023. DOI: 10.48550/arXiv.2306.06031. eprint: 2306.06031.

[47] Emanuele Rossi et al. "Temporal Graph Networks for Deep Learning on Dynamic Graphs". In: *ICML 2020 Workshop on Graph Representation Learning and Beyond*. 2020. DOI: 10.48550/arXiv.2006.10637.

[48] Da Xu et al. "Inductive representation learning on temporal graphs". In: *International Conference on Learning Representations (ICLR)*. 2020. DOI: 10.48550/arXiv.2002.07962.

[49] Yanbang Wang et al. *Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks*. 2022. DOI: 10.48550/arXiv.2101.05974.

[50] Michael Schlichtkrull et al. "Modeling Relational Data with Graph Convolutional Networks". In: *European Semantic Web Conference*. Springer. 2018, pp. 593–607. DOI: 10.1007/978-3-319-93417-4_38.

[51] Ziniu Hu et al. "Heterogeneous Graph Transformer". In: *Proceedings of The Web Conference 2020*. 2020, pp. 2704–2710. DOI: 10.1145/3366423.3380027.

[52] X. Wang et al. "Heterogeneous Graph Attention Network". In: *The World Wide Web Conference*. 2019, pp. 2022–2032. DOI: 10.1145/3308558.3313562.

[53] Zhitao Ying et al. "GNNExplainer: Generating Explanations for Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. DOI: 10.48550/arXiv.1903.03894.

[54] Ziqi Liu et al. *Heterogeneous Graph Neural Networks for Malicious Account Detection*. 2020. DOI: 10.48550/arXiv.2002.12307.

[55] Yaqin Zhou et al. "Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. DOI: 10.48550/arXiv.1909.03496.

[56] Narasimhan Jegadeesh and Joshua Livnat. "Revenue surprises and stock returns". In: *Journal of Accounting and Economics* 41.1-2 (2006), pp. 147–171. DOI: 10.1016/j.jacceco.2005.09.002.

[57] Wikipedia contributors. *List of S&P 500 companies*. Accessed: 2025-04-21. 2025. URL: https://en.wikipedia.org/wiki/List_of_S%26P_500_companies.

[58] Ran Aroussi. *yfinance: Yahoo! Finance market data downloader*. Accessed: 2025-04-22. 2019. URL: https://github.com/ranaroussi/yfinance.

[59] *SeleniumHQ Browser Automation*. https://www.selenium.dev/.

[60] Securities and Exchange Commission. "Securities and Exchange Commission EDGAR". In: (2025). URL: https://www.sec.gov/edgar/search/.

[61] Zihan Dong, Xinyu Fan, and Zhiyuan Peng. *FNSPID: A Comprehensive Financial News Dataset in Time Series (GitHub repository)*. Accessed: 2025-05-14. 2024. URL: https://github.com/Zdong104/FNSPID_Financial_News_Dataset.

[62] Finnhub. *Finnhub Stock API*. Accessed: 2025-08-10. URL: https://finnhub.io/.

[63] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017. DOI: 10.48550/arXiv.1609.02907.

[64] Petar Veličković et al. "Graph Attention Networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2018. DOI: 10.48550/arXiv.1710.10903.

[65] Diederik P. Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)* (2015). DOI: 10.48550/arXiv.1412.6980.

[66] Petar Veličković et al. "Graph Attention Networks". In: *International Conference on Learning Representations (ICLR)*. 2018. DOI: 10.48550/arXiv.1710.10903.

[67] Zhengxin Joseph Ye and Bjorn W. Schuller. *Capturing dynamics of post-earnings-announcement drift using genetic algorithm-optimised supervised learning*. 2020. DOI: 10.48550/arXiv.2009.03094. arXiv: 2009.03094 [q-fin.ST].

[68] Andy Chung and Kumiko Tanaka-Ishii. "Predictability of Post-Earnings Announcement Drift with Textual and Contextual Factors of Earnings Calls". In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. ICAIF '23. Brooklyn, NY, USA: Association for Computing Machinery, 2023, pp. 401–408. ISBN: 9798400702402. DOI: 10.1145/3604237.3626861.

A. Appendix

A.1 Data Collection and Preparation Visualisations

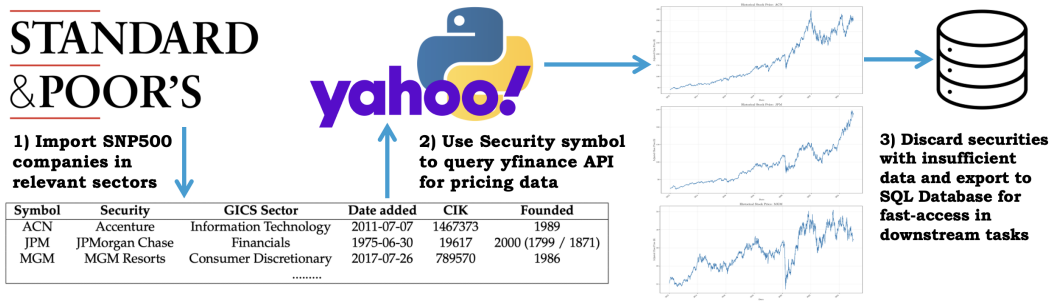


FIGURE A.1: Pipeline for extraction of company metadata and historical stock pricing

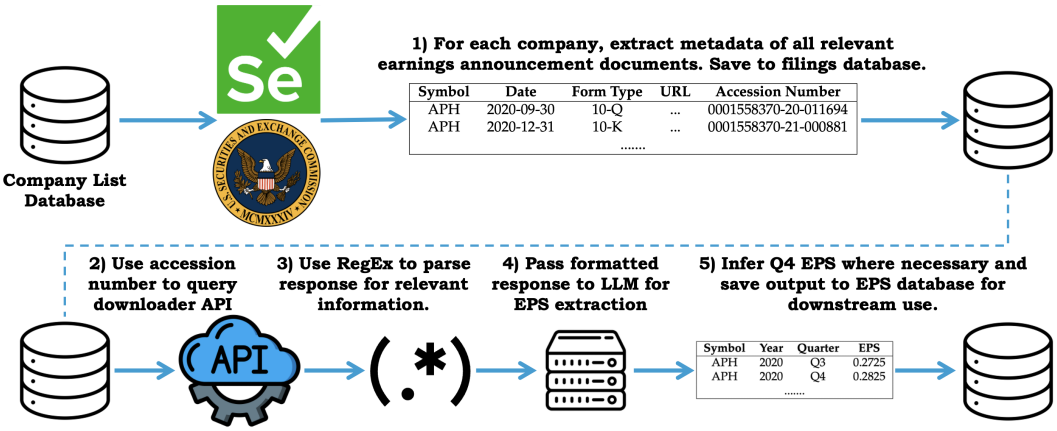


FIGURE A.2: Pipelines for extraction of filings metadata from SEC EDGAR and EPS extraction from raw HTML of filings.

A.2 Temporal Decay Functions Visualisation

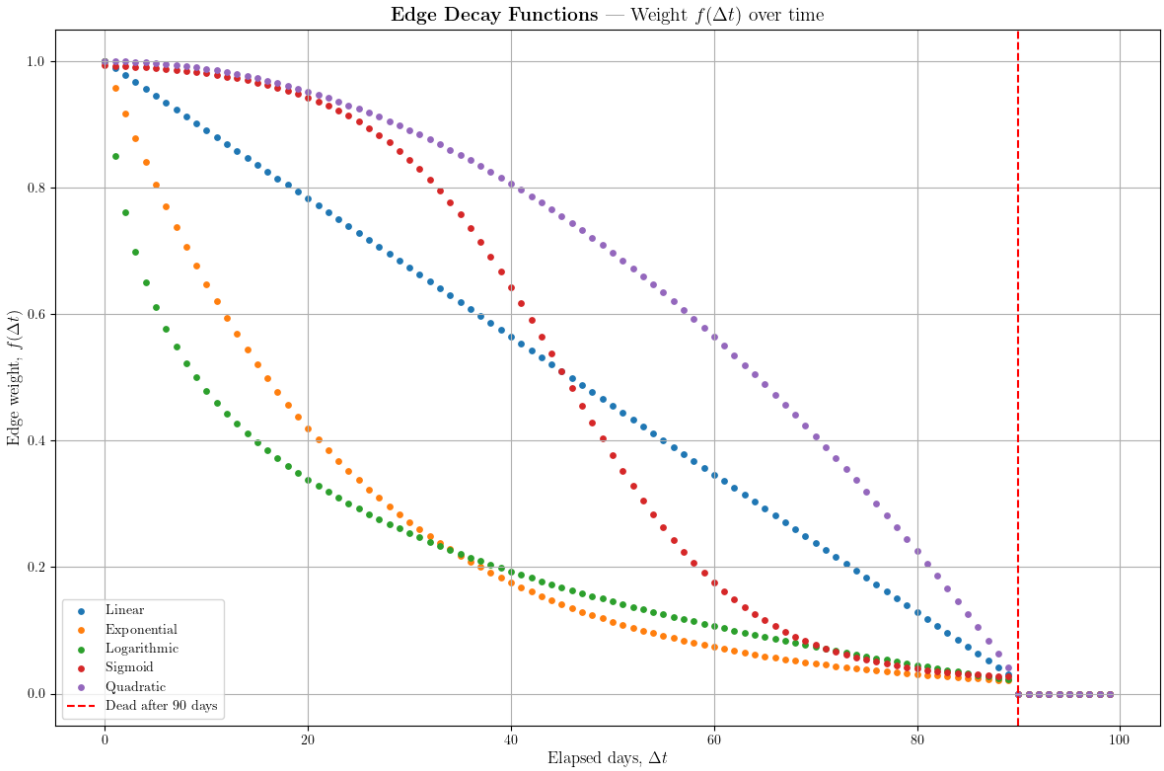


FIGURE A.3: Illustration of different temporal decay functions $\delta(\Delta t)$, showing how event influence diminishes with time lag Δt.

A.3 Financial News Dataset Exploratory Data Analysis

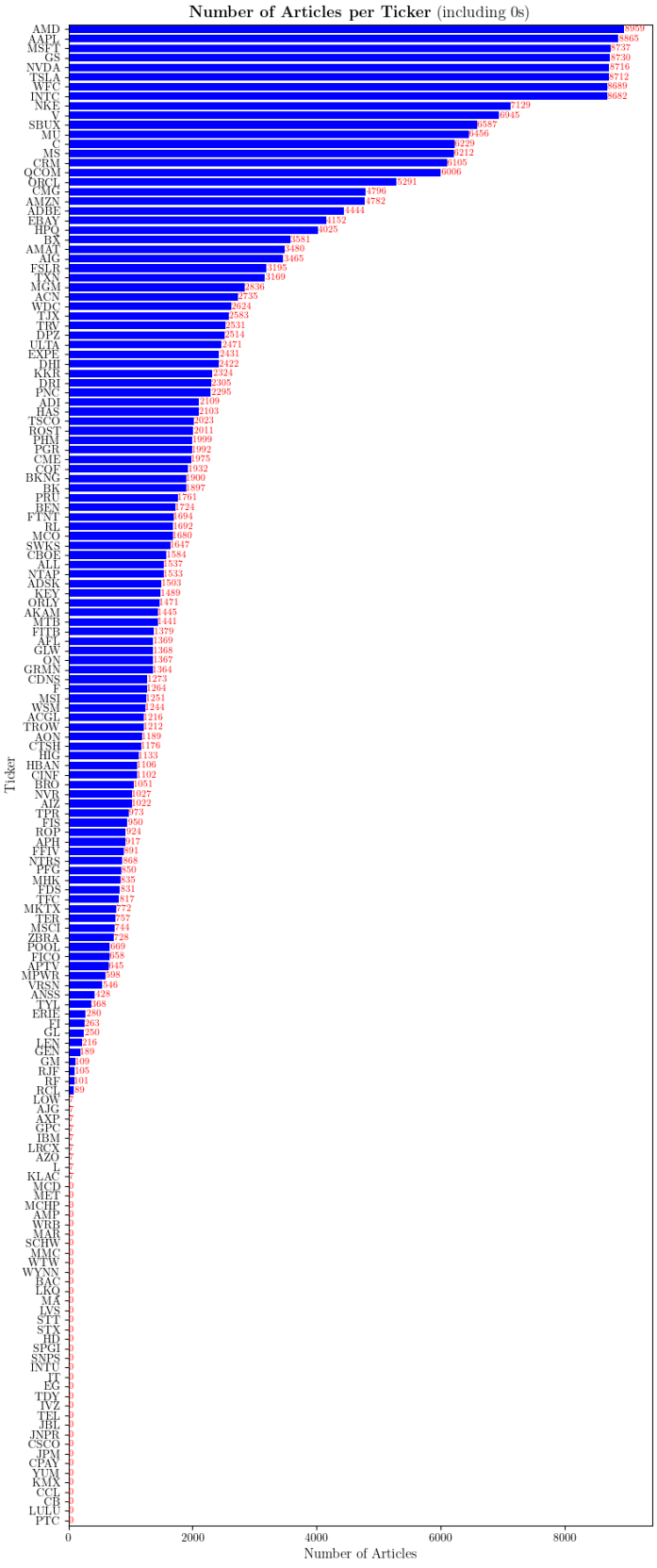


FIGURE A.4: Bar chart showing article count per primary ticker sorted by article count

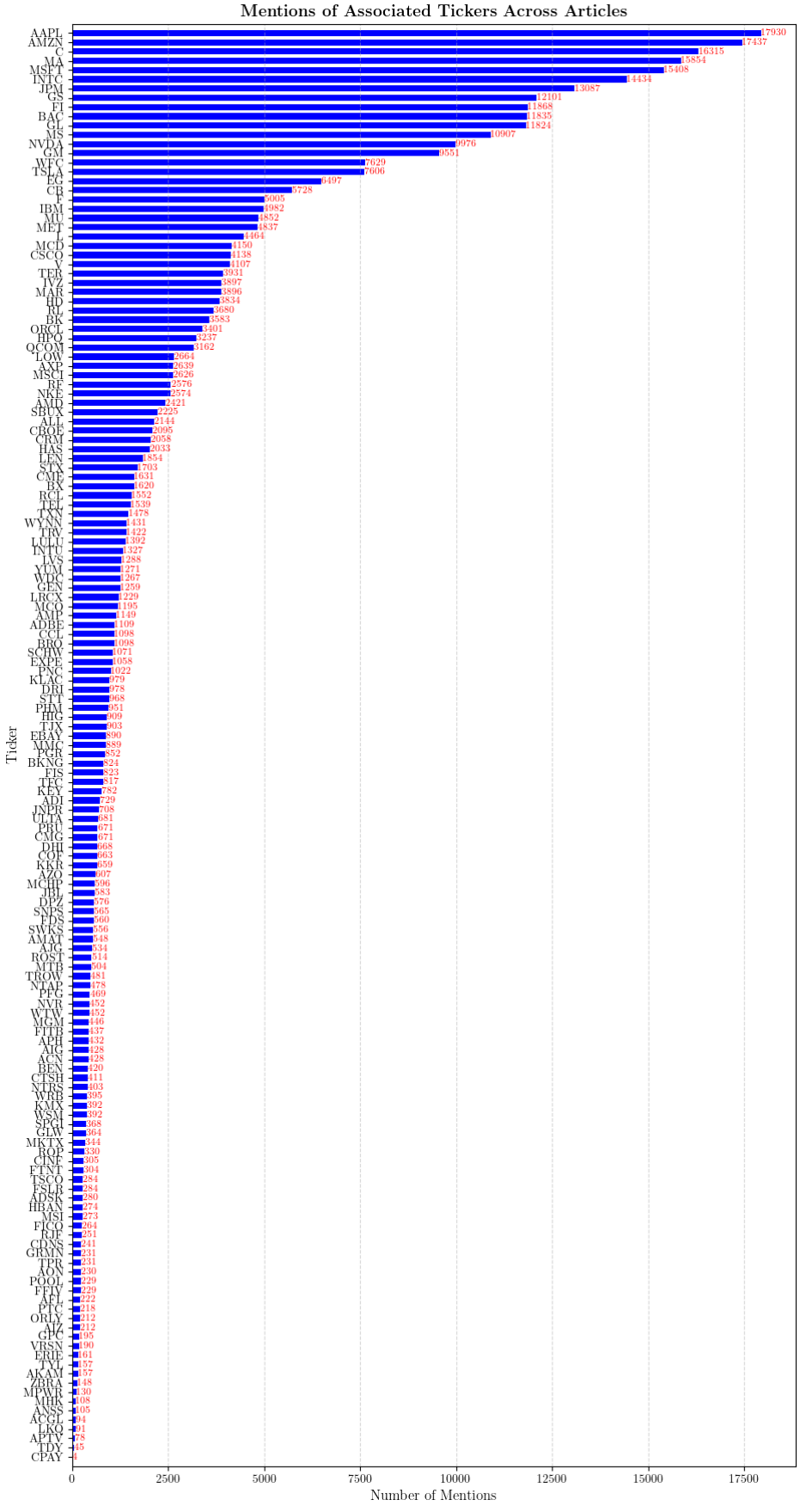


FIGURE A.5: Bar chart showing article count per assocaited ticker sorted by article count

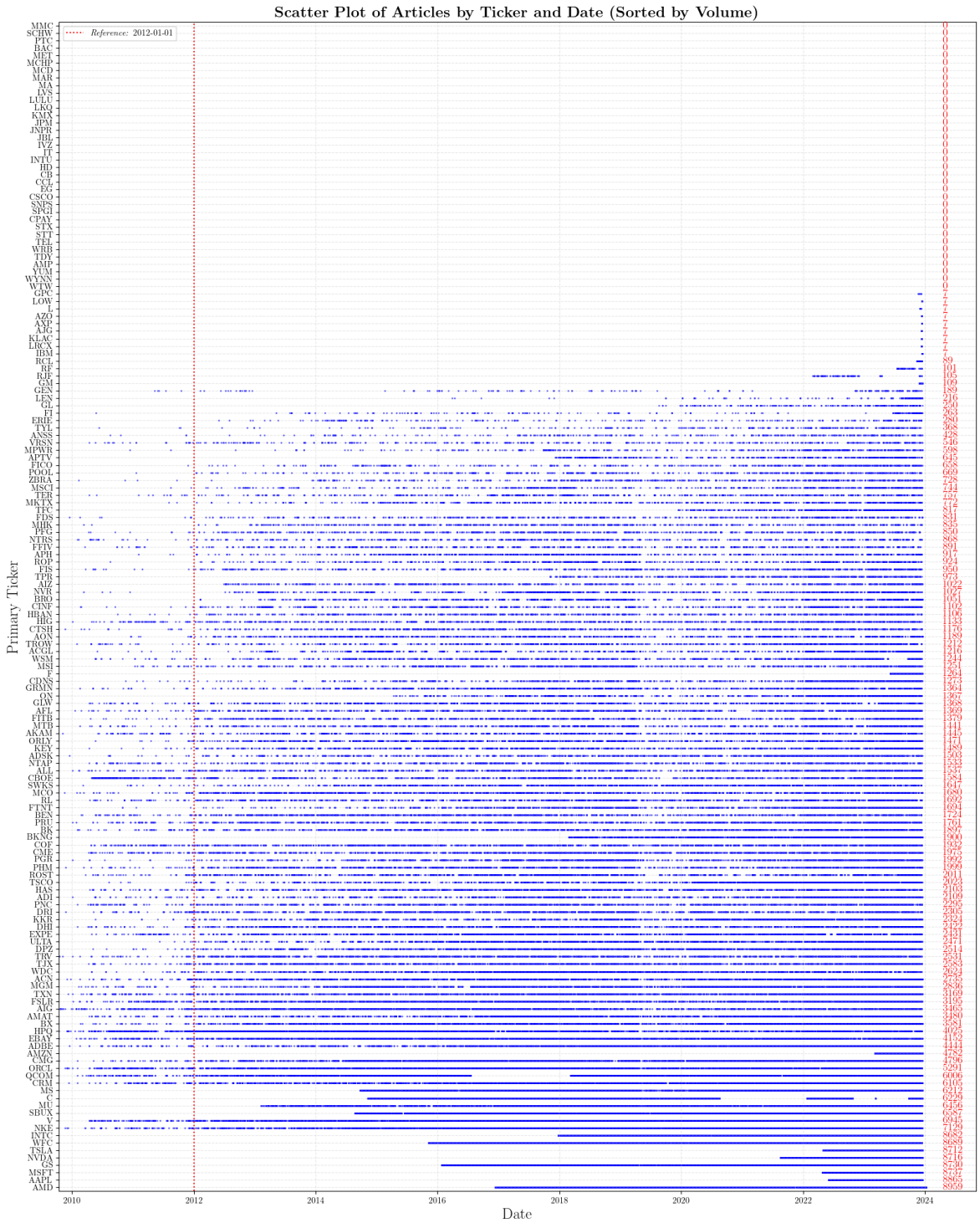


FIGURE A.6: A Scatter plot of articles by primary ticker over time, sorted by article volume.

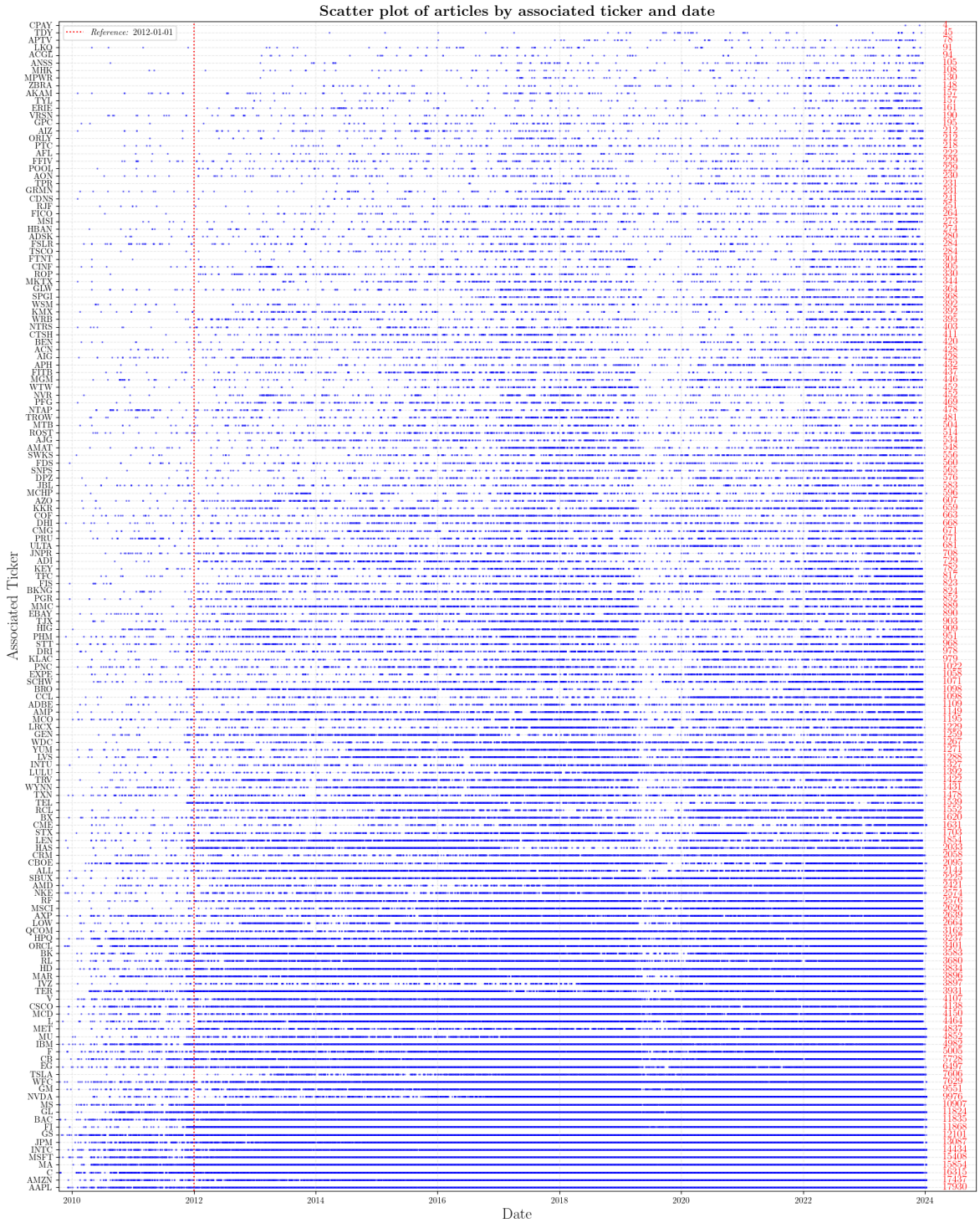


FIGURE A.7: A Scatter plot of articles by associated ticker over time, sorted by article volume.

A.4 Subgraph Visualisation

Figure A.8 shows the a visualised knowledge graph for news articles relating to eBay (EBAY) leading up to their 2021 Q3 earnings announcment (where PEAD did not occur). Figure A.9 shows the quantitative metrics embedded in the EBAY company node, and Figure A.10 shows the textual metadata embedded in a sample fact node.

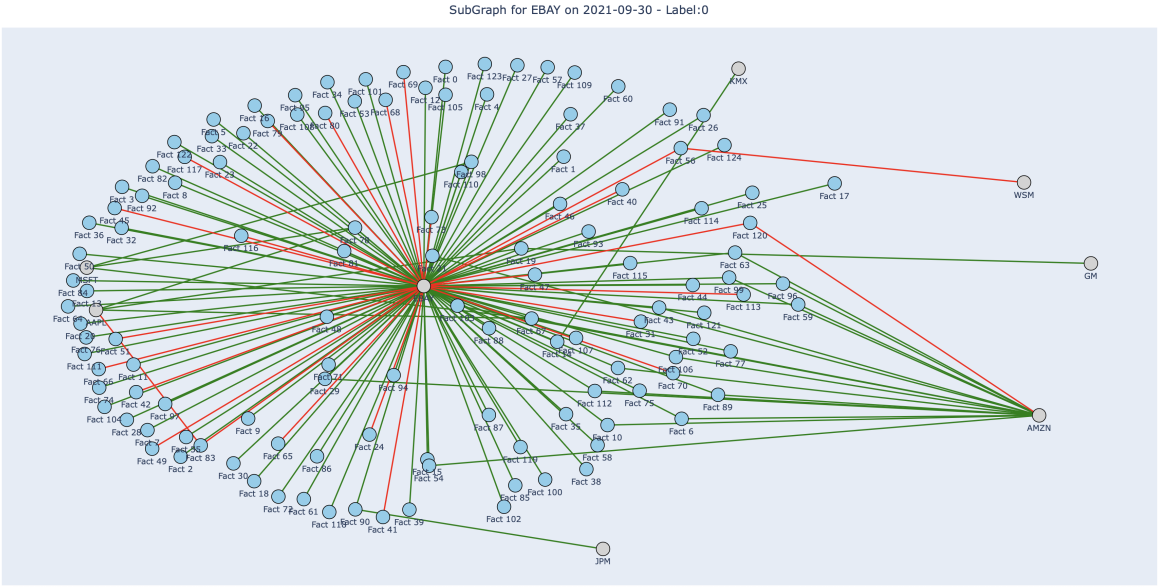


FIGURE A.8: Visualised knowledge subgraph for news relating to eBay (EBAY) prior to the 2021 Q3 earnings announcement

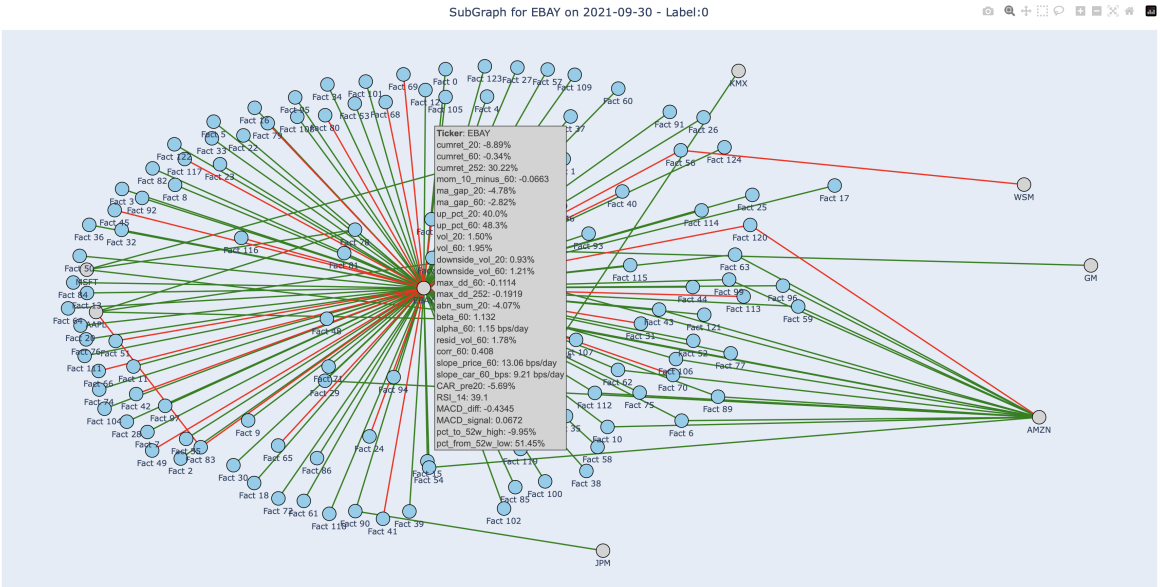


FIGURE A.9: Company-level view of the EBAY node, showing embedded quantitative metrics associated with the earnings event.

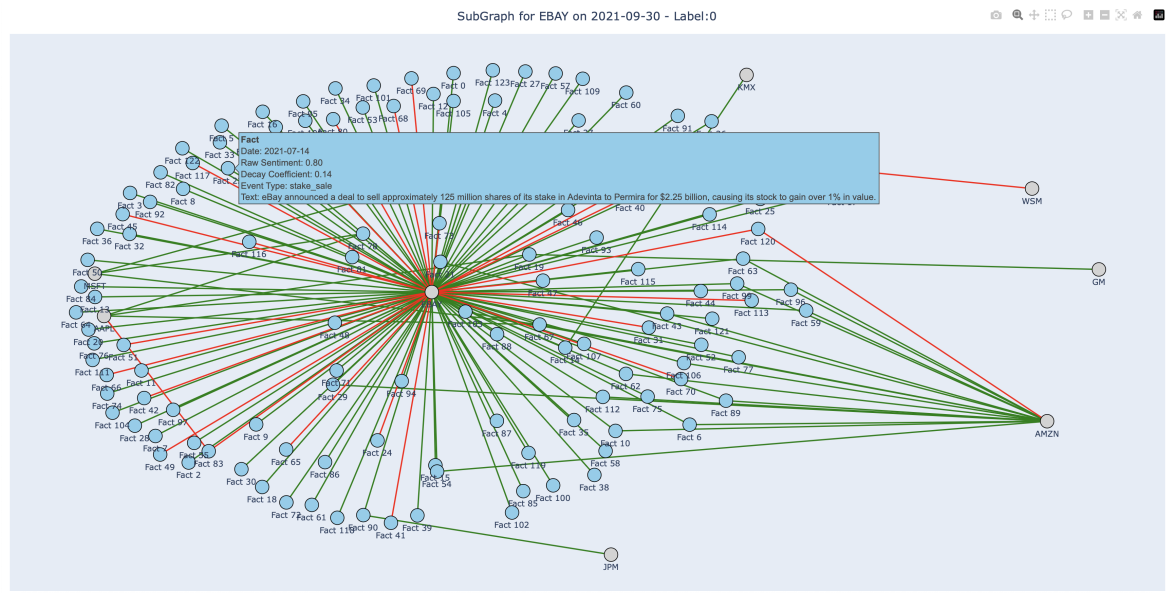


FIGURE A.10: Fact-level view of a sample node, showing textual metadata including event type, sentiment, and date and text summary.