Journal of Machine Learning Research 27 (2026) 1-37 Submitted 9/25; Revised 5/26; Published 6/26

AgentPEN: A Prediction-Explanation Network for Sequential Stock Movement via LLMs and Recurrent Generation

Shuqi Li* SHUQI.LI@KAUST.EDU.SA
Gaoling School of Artificial Intelligence, Renmin University of China, China
King Abdullah University of Science and Technology, Saudi Arabia

Mengyao Guo* GUOMY7299@RUC.EDU.CN
School of Information, Renmin University of China, China

Yunzhong Zheng YZ9712@NYU.EDU
New York University, USA

Siqi Li TSIKU0620@GMAIL.COM
Xi'an University of Finance and Economics, China

Xin Gao XIN.GAO@KAUST.EDU.SA
King Abdullah University of Science and Technology, Saudi Arabia

Rui Yan† RUIYAN@RUC.EDU.CN
School of Artificial Intelligence, Wuhan University, China

Editor: Christian Shelton

Abstract

The importance of explainability in stock prediction is increasingly recognized, especially for audit and regulatory purposes. Meanwhile, financial news corpora are often key drivers behind stock price fluctuations. However, the raw news data obtained is usually highly noisy, has a highly variable scope of influence in time and space, and is not precisely synchronized with stock price data. In this paper, we propose a prediction-explanation network called AgentPEN, which can provide clear explanations for complex temporal price patterns. Specifically, AgentPEN jointly aligns text and price streams by an LLM-based Representation Fusion Agent and then adopts a Deep Recurrent Generation module to explore the distribution of stock movements. The LLM-based Representation Fusion Agent is designed in a Selection-Memory-Fusion manner: the Text Selection Module picks up useful information from massive text data; the Text Memory Module evaluates and writes the text memory from a two-view perspective including Temporal Memory and Spatial Memory; the Information Fusion Module models the interaction between text and price data. Next, the fused representation is sent to the Deep Recurrent Generation module to convert insights into stock movement predictions. Experiments on multiple real-world datasets have shown that AgentPEN surpasses the state-of-the-art baselines both in prediction accuracy and explainability.

Keywords: time series prediction, large language model, financial technology, variational autoencoder, prediction explainability

*. These authors contributed equally.
†. Corresponding author.

©2026 Shuqi Li, Mengyao Guo, Yunzhong Zheng, Siqi Li, Xin Gao, Rui Yan.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v27/25-2175.html`.

1. Introduction

Artificial Intelligence (AI) has become a staple in the financial sector, addressing various challenges such as stock movement prediction (Luo et al., 2023; Xu and Cohen, 2018), stock recommendation (Gao et al., 2022; Coleman et al., 2022; Gao et al., 2021; Ma et al., 2022; Hsu et al., 2021), risk management (Ahbali et al., 2022; Wang et al., 2021b), factor mining (Yu et al., 2023; Ren et al., 2024), etc. In essence, accurate stock movement prediction is fundamental and can provide insights into other high-level tasks. Thus, stock price movement prediction has drawn enormous interest from both academia and industry (Frankel, 1995; Bollen et al., 2011; Edwards et al., 2018), and it is also the main task studied in this paper.

The Efficient Market Hypothesis (EMH) posits that stock prices reflect all available information, rendering it difficult to predict future price movements (Miller et al., 1970). However, subsequent research has highlighted limitations to this theory, emphasizing how phenomena like information asymmetry (Timmermann and Granger, 2004) and irrational behaviors (Lo, 2004) can lead to deviations from perfect market efficiency. These findings have spurred researchers to seek ways to achieve excess market returns by identifying and capitalizing on these inefficiencies (Wang et al., 2024). Building on this foundation, there is a burgeoning interest in exploring various data types to improve prediction capabilities (Wang et al., 2021a). Recent research has empirically demonstrated that financial news commentaries and social media posts reveal rich information about the market (Bollen et al., 2011; Li et al., 2014; Schumaker and Chen, 2009). This has led to the emergence of a new field called news-driven stock movement prediction. By leveraging the financial texts from diverse data sources, researchers could establish more robust methods while improving the explainability of methods, ultimately contributing to more informed investment decisions.

Even though recent works have significantly improved stock movement prediction performance by modeling textual data, we still observe the following challenges in financial textual data modeling: 1) Massive noise exists in financial text data, and irrelevant or ambiguous texts should be filtered out (Sprenger and Welpe, 2001; Duan et al., 2025). 2) The impact of each text on stocks may vary in time and space. For instance, the influence of policy-related texts may last longer and have a broader scope compared to entertainment-related texts. 3) Most current methods that handle both textual data and stock price data simply concatenate the embeddings of the two data types, lacking precise alignment and an interaction process.

Traditional text encoding methods, such as BERT and RoBERTa (Devlin et al., 2018; Liu et al., 2019), are insufficient to fully address these challenges. In contrast, with the development of Natural Language Processing (NLP) techniques, Large Language Models (LLMs) have unique advantages to process text data in the above financial situations due to their advanced knowledge and reasoning abilities (Roy and Roth, 2016; Bubeck et al., 2023; Xiao et al., 2024).

With these motivations in mind, in this paper, we propose a novel news-driven stock movement prediction-explanation framework called AgentPEN to jointly predict the stock price movement and explore probable effect factors from text data. The key component of the proposed AgentPEN is an LLM-based Representation Fusion Agent which is designed in a Selection-Memory-Fusion manner: first, select stock-related news texts; next, store and organize the texts with temporal-level memory and spatial-level memory respectively; at last, integrate the two modalities, i.e. textual news data and numerical price data, into the fusion representations. After processing market information, to more accurately capture the stochastic elements of the stock market, we model the distribution of output movement space using a generative model conditioned on the input market data and then leverage a temporal attention mechanism to predict the target movements enhanced by the former predicted movements.

To sum up, the contributions of our work include:

- A novel framework called AgentPEN for news-driven stock movement prediction with better explainability through alignment of news text time series data and the price numerical time series data.
- An LLM-based Representation Fusion Agent designed in a Selection-Memory-Fusion manner to integrate the multi-modal market information.
- Collection and release of a new dataset for the U.S. stock market, which includes comprehensive financial news texts and historical stock price data. The dataset is publicly available at https://github.com/Shuqi-li/AgentPEN-US, facilitating further research in the Fintech field.

We carry out comprehensive tests on real-world datasets to validate the performance of our proposed AgentPEN. Experiments on three distinct datasets covering different periods illustrate that AgentPEN surpasses current state-of-the-art models, proving that the LLM-based Representation Fusion Agent can significantly enhance prediction accuracy and explainability.

2. Related Work

In the realm of stock trading practices, stock movement prediction tasks are primarily guided by two prevalent analytical methods: technical analysis and fundamental analysis.

2.1 Technical Analysis-based Stock Movement Prediction

Technical analysis is favored by investors and traders who predict stock prices based on historical price patterns and market behavior. They believe that the historical price patterns will reappear in the future.

Since stock data is a series of data that changes over time, time series prediction methods, especially RNN, are applied to this field. ALSTM (Qin et al., 2017) captured the most relevant input features through dual attention, thus achieving the goal of long-term dependent sequence prediction. Adv-ALSTM (Feng et al., 2019) improved the robustness of the prediction model by adding perturbations to high-level features and introducing adversarial training. In recent years, due to transformers' powerful representation and long sequence processing capabilities, attention mechanisms have been widely applied in the field of time series prediction. DTML (Yoo et al., 2021) combined transformer and LSTM to model end-to-end asymmetric correlation between stocks.

2.2 Fundamental Analysis based Stock Movement Prediction

On the other hand, fundamental analysis goes beyond mere price trends to evaluate the intrinsic value of a stock. This method considers a variety of factors apart from historical prices, including detailed financial statements, industry trends, economic conditions, and even broader market sentiments. By integrating these diverse elements, fundamental analysis provides a comprehensive view of a stock's potential, offering a robust basis for making informed investment decisions.

With the rapid development of NLP techniques, text from social media and online news has become a newly popular source of fundamental analysis. HAN (Hu et al., 2018) simulated the mechanism of human beings processing news, proposed a hybrid attention model, and handled the impact of different quality Internet news data on stock predictions. Stocknet (Xu and Cohen, 2018) introduced recurrent, continuous latent variables to better handle the randomness of stock text information. MSHAN (Gong and Eldardiry, 2021) proposed to use a multi-stage TCN-LSTM method to capture unstructured text data information and long-range text dependencies. PEN (Li et al., 2023) applied a shared representation learning module to capture the correlation between text data and stock prices, which achieved prediction and interpretation of stock price movements. Additionally, many works modeled the correlation between stocks to enhance stock price prediction. MAN-SF (Sawhney et al., 2020) applied graph neural networks to process complex time signals of financial data, social media, and stock relationships in a hierarchical temporal manner for stock movement prediction. CMIN (Luo et al., 2023) avoided attention to false stock correlations by calculating the transfer entropy between multivariate stocks and simultaneously introducing a fusion mechanism to learn the interactive effects in the information flow. Co-CPC (Wang et al., 2021a) proposed a deep model based on copula, which reduces model uncertainty and data uncertainty by modeling macroeconomic characteristics and stock movements.

2.3 Prediction Explainability

Despite significant progress in the development of deep stock price prediction models, their practical adoption remains disappointingly scarce, largely attributed to their opaque, "black box" nature. The inner workings of these models are often not transparent, making it difficult for users to understand how predictions are derived. Consequently, there is a growing concern regarding the trustworthiness and reliability of such systems (Chen and Wang, 2025). To date, only a handful of research studies have tackled the issue of prediction explainability, aiming to demystify the processes behind these advanced models and enhance their accessibility and user trust. Hu et al. (2018) imitated the learning process of human beings facing chaotic online news, driven by three principles: sequential content dependency, diverse influence, and effective and efficient learning. Meanwhile, they designed a Hybrid Attention Network (HAN) to predict the stock trend based on the sequence of recent related news and leveraged attention weights to explain the news potentially causing stock price fluctuations. To timely select the textual stories associated with that price time series, Dang et al. (2019) proposed a multi-modal neural model called MSIN. By multiple steps of data interrelation between the textual data and the numerical data, MSIN learned to focus on a small subset of text articles that best align with the performance in the time series and could explain and enhance the stock prediction results. After that, Li et al. (2023) proposed a

Shared Representation Learning (SRL) module to learn the shared representation between text and price data, which generated a Vector of Salience (VoS) for every sample to explain the importance of individual text documents in the corpora. With the development of Large Language Models (LLMs), LLM-based agent technologies have also rapidly advanced, even in finance. Tong et al. (2024) proposed a financial LLM framework that consists of PloutosGen and PloutosGPT. PloutosGen contains multiple LLM-based experts capable of analyzing diverse modal data to deliver quantitative strategies from varied angles. PloutosGPT then synthesizes these insights and predictions, producing coherent and interpretable rationales. Wang et al. (2024) designed Sequential Knowledge-Guided Prompting (SKGP) to identify factors that influence stock movements, which focuses on mining the factors that directly influence stock market changes, providing clear explanations for complex temporal variations. K. J. Koa et al. (2024) proposed Summarize-Explain-Predict (SEP), which utilizes self-reflective agents and PPO to enable LLMs to autonomously learn and generate explainable stock predictions without the need for manual annotation.

3. Problem Description & Architecture

In this paper, we focus on tackling the news-driven stock movement prediction task. In this section, we introduce the task description and the architecture of the designed network.

3.1 Problem Description

Assuming the look-back window is set as L, we leverage the historical market information $\boldsymbol{X} = \{x_t\}_{t=T-L+1}^{T}$, where each x_t includes news texts c_t and price data p_t to predict the movement of adjusted close prices y_T on trading day T.

For the network input information, we denote the news text set as $\boldsymbol{C} = \{c_t\}_{t=T-L+1}^{T}$, where c_t includes M pieces of news texts on time step t, $c_t = \left[c_t^1, c_t^2, \ldots, c_t^m, \ldots, c_t^M\right]$. For the price data, we leverage the same price features with the previous works (Xu and Cohen, 2018; Li et al., 2023; Hu et al., 2018), $\tilde{p}_t = \left[\tilde{p}_t^c, \tilde{p}_t^h, \tilde{p}_t^l\right] \in \mathbb{R}^{1\times 3}$ consists of adjusted close price $\tilde{p}_t^c$, high price $\tilde{p}_t^h$ and low price $\tilde{p}_t^l$. To align the scale of price features, we normalize them with the last adjusted closing prices $p_t = \tilde{p}_t / \tilde{p}_{t-1}^c - 1$. Then the historical price input is denoted as $\boldsymbol{P} = \{p_t\}_{t=T-L+1}^{T}$.

For the network prediction output, the movement could be constructed as a binary variable:

$$y_T = \begin{cases} 1, \ p_T^c < p_{T+1}^c \\ 0, \ p_T^c \geq p_{T+1}^c, \end{cases} \tag{1}$$

where $y_T = 1$ indicates that the adjusted close price rises on the next trading day, and $y_T = 0$ otherwise. Besides, we set the adjusted close price as the movement indicator because it reflects the stock's value after accounting for any corporate actions (Yoo et al., 2021).

In a nutshell, this paper aims to predict the movement y_T based on $\boldsymbol{X} = \{x_t\}_{t=T-L+1}^{T} = \{c_t, p_t\}_{t=T-L+1}^{T}$.

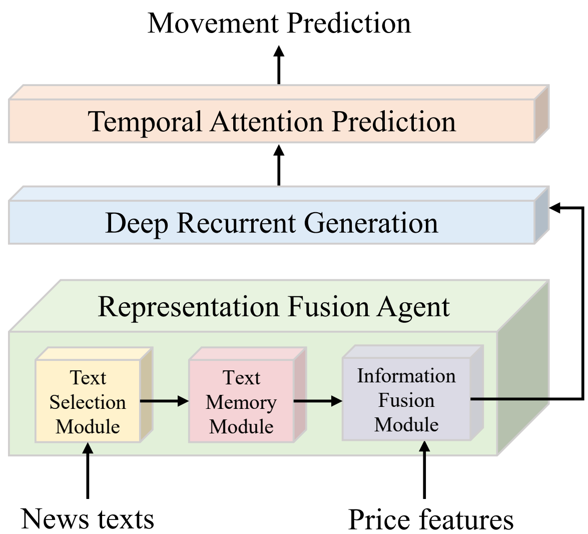

Figure 1: The overall architecture of the proposed stock Prediction-Explanation Network, including the Representation Fusion Agent, the Deep Recurrent Generation Module and the Temporal Attention Prediction Module.

3.2 Model Architecture

The overall architecture of AgentPEN is as shown in Figure 1. It comprises three main components:

- **An LLM-based Representation Fusion Agent (RFA).** It is designed to efficiently analyze and integrate both textual $\boldsymbol{C}$ and numerical data $\boldsymbol{P}$. This integration occurs through the orchestration of three specialized modules: **Text Selection Module** filtering out irrelevant historical news data to reduce noise and ensure that only pertinent information will be fed into the next module; **Text Memory Module** taking the selected text from the prior module and memorizing the texts into the Temporal & Spatial Memory; **Information Fusion Module** integrating the textual and numerical data streams into an overall representation which offers a holistic view of the market conditions, leading to more robust and accurate prediction performance.

- **Deep Recurrent Generation (DRG).** We leverage a conditional variational autoencoder that describes the potential movement distribution in latent space with a probabilistic manner. It receives the fused information from the Representation Fusion Agent, first infers the latent variable set $\boldsymbol{Z}$ and then decodes movements $\boldsymbol{y}$ for every time step in the look-back window.

- **Temporal Attention Prediction (TAP).** We adopt an attention mechanism to explore the relationship between the movement of the target trading day y_T and its former movements $y_{<T}$. This needs the network to decode not only the movement of the target trading day but also previous movements. Then we could utilize the past movement labels for training, thereby further enhancing the prediction performance.

4. Modeling Details of AgentPEN

We aim to predict the stock's movement with historical market information as input. In this section, we introduce the main modules of AgentPEN in detail: LLM-based Representation Fusion Agent for information filtering, aligning, and fusing; the Deep Recurrent Generation module for encoding the latent variable and then decoding movements; and the Temporal Attention Prediction module for the final prediction.

4.1 LLM-based Agent for Representation Fusion

This architecture is engineered to meticulously analyze and amalgamate textual c_t and numerical p_t data within the domain of financial quantification. This synthesis is facilitated via the orchestration of three distinct modules: the Text Selection Module, the Text Memory Module, and the Information Fusion Module. Figure 2 shows the whole pipeline of this part. The formal algorithm of the LLM-based Representation Fusion Agent is shown in Algorithm 1 in Appendix A. Moreover, the prompts used for $LLM_{\text{selection}}$, LLM_{temporal} and LLM_{spatial} are shown in Table 8 in Appendix B.

4.1.1 Text Selection Module

As defined in Section 3.1, the input news text set is denoted as $\boldsymbol{C} = \{c_t\}_{t=T-L+1}^{T}$, where c_t includes M pieces of news texts on time step t, $c_t = \left[c_t^1, c_t^2, \dots, c_t^m, \dots, c_t^M\right]$. Based on the input texts, the Text Selection Module, as shown in the left part of Figure 2, prompts an $LLM_{\text{selection}}$ to pick up the salient information (correlation score more than the correlation threshold τ_c) from extensive news texts. It then outputs the filtered news corpora $\boldsymbol{C^s} = \{c_t^s\}_{t=T-L+1}^{T}$, where c_t^s includes S pieces of filtered news texts on time step t, $c_t^s = \left[c_t^1, c_t^2, \dots, c_t^S\right]$. These selected news will be fed into the next Text Memory Module.

This module offers several distinct advantages for enhancing stock price predictions. First, by utilizing a sophisticated prompt mechanism, the module selectively extracts the relevant news from extensive noisy market information. This targeted approach ensures that only the most relevant data is processed, significantly reducing noisy information.

4.1.2 Text Memory Module with Temporal-Spatial view

This section introduces the Text Memory Module, as shown in the middle part of Figure 2, which is designed to enhance stock price prediction models by leveraging both the temporal and spatial memory mechanism: Temporal Memory is designed to overcome the limitations of traditional time-lagged models by providing a more flexible and comprehensive approach to the varied impact duration of news. Meanwhile, Spatial Memory addresses the interconnectedness of the stock market by considering how news of one stock can influence others. This component is crucial for understanding market sentiment and industry trends that affect multiple stocks. Based on the written memories, the memory retrieval part selectively retrieves news, thus refining the useful information.

Temporal Memory. Temporal memory incorporates a dynamic time window mechanism, enabling the agent to operate on a wider range of events over a longer retrieval period. We prompt the LLM to evaluate the duration of each news text's impact on the current stock, which involves assessing the significance of the news and the time lag of its

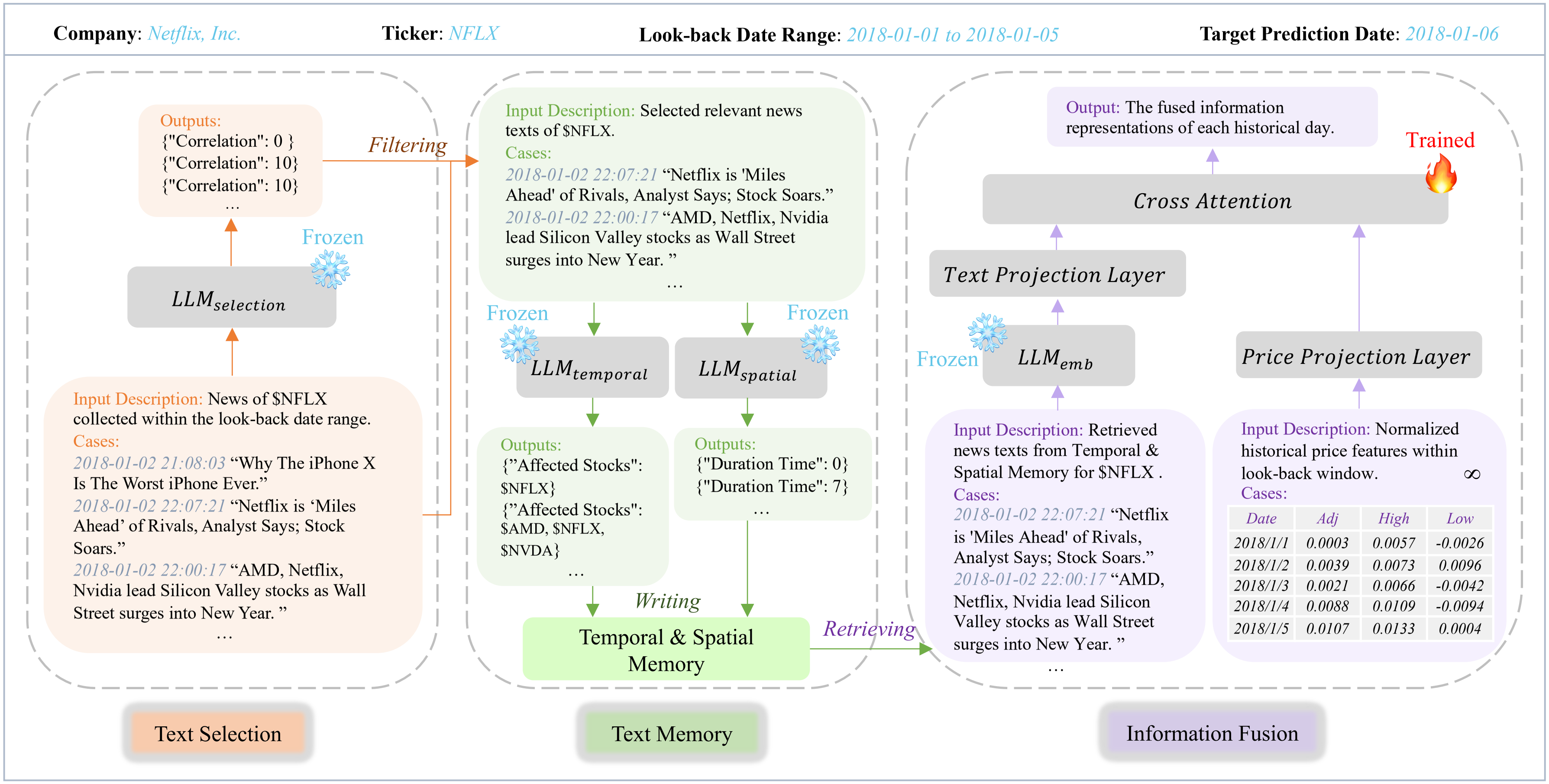

Figure 2: The pipeline of LLM-based Representation Fusion Agent (RFA), which includes three modules: **Text Selection Module**, **Text Memory Module**, and **Information Fusion Module**.

impact. Formally, for the filtered news corpora $\boldsymbol{C^s} = \{c_t^s\}_{t=T-L+1}^{T}$, the temporal memory prompts an LLM_{temporal} and then outputs the news impact duration $\boldsymbol{D} = \{d_t\}_{t=T-L+1}^{T}$, where $d_t = \left[d_t^1, d_t^2, \ldots, d_t^S\right]$ is a duration time list of news on time step t. Then we could write the news and its impact duration time into the temporal memory. This allows the model to incorporate the influence of past news from days to months, reflecting the real-world impact duration of different news types.

Spatial Memory. Spatial memory implicitly constructs a network map of stocks that are commonly affected by similar news sources. This mapping allows the model to identify potential indirect impacts on the target stock from developments in related stocks or sectors. We prompt the LLM to analyze the potential impact range of each news text, i.e. the stocks that may be affected. Formally, for the filtered news corpora $\boldsymbol{C^s} = \{c_t^s\}_{t=T-L+1}^{T}$, the spatial memory prompts an LLM_{spatial} and then outputs the news impact range $\boldsymbol{R} = \{r_t\}_{t=T-L+1}^{T}$, where $r_t = \left[r_t^1, r_t^2, \ldots, r_t^S\right]$, r_t is a list of stock tickers that may be affected by this news text. Then we will write the news and its impact range list into the spatial memory.

Memory Retrieval. After the memory writing, we design a memory retrieval process to retrieve news from both temporal and spatial memories selectively.

Upon receiving a target stock and target prediction trading date, the memory retrieval module retrieves the top-K pivotal memory news. These pieces of news are chosen according to their impact duration and impact range, which are generated by the evaluation results in the memory writing stage. Specifically, for each target stock and date, we will scan the temporal and spatial memory to find news items whose effect duration includes the target date and to find the news that has a potential impact on the target stock. The output of memory retrieval stage could be denoted as $\boldsymbol{C^k} = \left[c^1, c^2, \ldots, c^K\right]$, including K pieces of retrieved news texts.

The synergy between temporal and spatial memory allows the text memory module to produce a holistic view of the news landscape's impact on stock prices, factoring in both direct impacts and the broader market context.

4.1.3 Information Fusion Module

In this part, the Information Fusion Module is designed to further integrate prices with news through attention mechanisms, as shown in the right part of Figure 2.

The inputs of this module are the retrieved news text $\boldsymbol{C^k} = \left[c^1, c^2, \ldots, c^K\right]$ and the historical price $\boldsymbol{P} = \{p_t\}_{t=T-L+1}^{T}$. Firstly, we leverage a text embedding LLM LLM_{emb} to encode the news $\boldsymbol{C^k}$ and then leverage a feature mapping layer to obtain the final text embedding,

$$\boldsymbol{C^l} = LLM_{emb}(\boldsymbol{C^k}), \boldsymbol{C^e} = \boldsymbol{C^l} W_t + b_t, \tag{2}$$

$\boldsymbol{C^e} = \{c^e\}_{e=1}^{K} \in \mathbb{R}^{K \times d}$, where d is the text embedding dimension, W_t is the weight matrix and b_t is the bias vector.

Similarly, we employ a feature mapping layer to project price data $\boldsymbol{P}$ into a latent space of dimension d:

$$\boldsymbol{P^e} = \boldsymbol{P} W_p + b_p, \tag{3}$$

$\boldsymbol{P^e} = \{p_t^e\}_{t=T-L+1}^{T} \in \mathbb{R}^{L \times d}$, where W_p is the weight matrix and b_p is the bias vector.

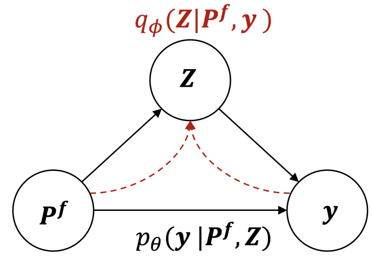

Figure 3: The graphic illustration of AgentPEN. AgentPEN leverages the fused market information representation $\boldsymbol{P^f}$ to infer the latent variable set $\boldsymbol{Z}$ and then predict the stock movement y. The forecasting process $p_\theta\left(\boldsymbol{y} \mid \boldsymbol{P^f}, \boldsymbol{Z}\right)$ is drawn with solid black lines with parameters θ and the variational inference process is denoted by dashed red lines with approximation parameters ϕ, $q_\phi\left(\boldsymbol{Z} \mid \boldsymbol{P^f}, \boldsymbol{y}\right)$ denotes the posterior distribution of the learned latent variable $\boldsymbol{Z}$.

Secondly, we design a cross attention module where each day's price embedding acts as the query, and each text embedding acts as the key and value. Formally, the attention process could be denoted as,

$$\boldsymbol{P^f} = \text{MultiheadAtt}\left(\boldsymbol{P^e}, \boldsymbol{C^e}\right), \tag{4}$$

where $\boldsymbol{P^f} = [p^f_{T-L+1}, \cdots, p^f_T] \in \mathbb{R}^{L \times d_{model}}$ is the output of the last attention layer, d_{model} is the hidden size. By allowing the price to attend to the news, we can extract information from the news that is relevant to price movements, enrich the original price embedding, and thus achieve information alignment and fusion $\boldsymbol{P^f}$. The fusion representations $\boldsymbol{P^f}$ are used as input to the Deep Recurrent Generation Module in the next section.

4.2 Deep Recurrent Generation

4.2.1 Overview

In this section, we model the distribution of stock movements conditioned on the fused market information representation, denoted as $p_\theta\left(y_T \mid \boldsymbol{P}^f\right)$ with parameters θ. This probabilistic formulation is motivated by the inherent stochasticity of stock movement prediction. Even under similar observed price patterns and news contexts, future movements may differ due to unobserved market factors, heterogeneous investor reactions, and external shocks. Therefore, directly learning a deterministic mapping from the fused representation $\boldsymbol{P}^f$ to the movement label may be overly restrictive.

To capture such uncertainty, we employ a conditional variational auto-encoder as the generative model, building upon recent developments in variational inference and directed graphical models (Xu and Cohen, 2018; Sohn et al., 2015; Li et al., 2023). The conditional VAE introduces latent variables $\boldsymbol{Z} = \{z_t\}_{t=T-L+1}^{T}$, which are modulated by the fused market information $\boldsymbol{P}^f$ and used to model the conditional movement distribution. Compared with a purely deterministic discriminative classifier, this design provides a principled way to incorporate stochastic latent variables into the sequential prediction process.

Formally, the conditional movement distribution can be written as

$$p_\theta\left(y_T \mid \boldsymbol{P}^f\right) = \int_{\boldsymbol{Z}} p_\theta\left(y_T, \boldsymbol{Z} \mid \boldsymbol{P}^f\right) d\boldsymbol{Z},$$

where the latent variables $\boldsymbol{Z}$ capture stochastic factors that are not fully observed in the input market information. The graphical illustration of this generative process is shown in Figure 3.

Inspired by Xu and Cohen (2018) and the multi-task learning strategy, we observe that predicting other lagged time steps within the look-back window can enhance the predictive performance of the target movement. Thus, in the training stage of AgentPEN, we make movement predictions not only for the target trading day T but also for the former movements existing in the look-back window. In other words, we aim to predict $\boldsymbol{y} = \{y_t\}_{t=T-L+1}^{T} = [y_{T-L+1}, \cdots, y_T]$ instead of just leveraging y_T as training label. Based on this observation, the prediction process could be factorized as follows:

$$\begin{aligned}
& p_\theta(\boldsymbol{y} \mid \boldsymbol{P^f}) \\
&= \int_{\boldsymbol{Z}} p_\theta(\boldsymbol{y}, \boldsymbol{Z} \mid \boldsymbol{P^f}) \\
&= \int_{\boldsymbol{Z}} p_\theta\left(\boldsymbol{y} \mid \boldsymbol{P^f}, \boldsymbol{Z}\right) p_\theta\left(\boldsymbol{Z} \mid \boldsymbol{P^f}\right) \\
&= \int_{\boldsymbol{Z}} p_\theta\left(y_T \mid p^f_{\le T}, z_T\right) p_\theta\left(z_T \mid z_{<T}, p^f_{\le T}\right) \prod_{t=T-L+1}^{T-1} p_\theta\left(y_t \mid p^f_{\le t}, z_t\right) p_\theta\left(z_t \mid z_{<t}, p^f_{\le t}, y_t\right),
\end{aligned} \tag{5}$$

where $p^f_{\le *} = [p^f_{T-L+1}, \cdots, p^f_{*}]$ and $p^f_{<*} = [p^f_{T-L+1}, \cdots, p^f_{*-1}]$ and the same definitions for z.

In this process, the prior of latent variable z is denoted as $p_\theta\left(z_t \mid z_{<t}, p^f_{\le t}\right), t \in [T - L + 1, T]$, which is modulated by the input market information. Since $y_{<T} = \{y_t\}_{T-L+1}^{T-1}$ is already known when predicting y_T, so it could be used in approximating the posterior $p_\theta\left(z_t \mid z_{<t}, p^f_{\le t}, y_t\right)$ in which $t \in [T - L + 1, T - 1]$.

We recurrently leverage the conditional variational auto-encoder to model Equation 5. This module could be divided into a Recurrent Variational Encoder and a Recurrent Variational Decoder and the illustration is shown in the blue part of Figure 4.

4.2.2 Recurrent Variational Encoder

We leverage deep neural networks to fit the distributions of latent variables $\boldsymbol{Z} = \{z_t\}_{t=T-L+1}^{T}$. Formally, we need to model the conditional prior distribution

$$p_\theta\left(\boldsymbol{Z} \mid \boldsymbol{P^f}\right) = \prod_{t=T-L+1}^{T} p_\theta\left(z_t \mid z_{<t}, p^f_{\le t}\right), \tag{6}$$

and the conditional posterior distribution

$$p_\theta\left(\boldsymbol{Z} \mid \boldsymbol{P^f}, \boldsymbol{y}\right) = \prod_{t=T-L+1}^{T-1} p_\theta\left(z_t \mid z_{<t}, p^f_{\le t}, y_t\right). \tag{7}$$

Unfortunately, the posterior distributions are all intractable. Following the key idea of VAE, we leverage the variational inference and reparameterization technique (Rezende et al., 2014; Jordan et al., 1999; Kingma and Welling, 2013) to approximate the posteriors by variational approximators $q_\phi\left(z_t \mid z_{<t}, p^f_{\leq t}, y_t\right)$, in which ϕ is the parameter set of approximations. The deep recurrent encoder is designed to infer the hidden states of latent variables recurrently. More specifically, we employ GRU as the basic recurrent framework to link the information from every time step.

We assume that all the prior and the posterior of latent variables $\{z_t\}_{t=T-L+1}^T$ follow standard multivariate Gaussian distributions:

$$\begin{aligned} p_\theta\left(z_t \mid z_{<t}, p^f_{\leq t}\right) &\sim \mathcal{N}\left(z_t; \mu_\theta, \sigma_\theta^2 I\right), \\ q_\phi\left(z_t \mid z_{<t}, p^f_{\leq t}, y_t\right) &\sim \mathcal{N}\left(z_t; \mu_\phi, \sigma_\phi^2 I\right). \end{aligned} \tag{8}$$

For the posterior, denoting the hidden state of encoder GRU as $h_t^{enc} = \mathrm{GRU}\left(p_t^f, h_{t-1}^{enc}\right)$. Then the hidden state of z_t could be calculated by,

$$h_t^{z^\phi} = \tanh\left(W_z^\phi\left[z_{t-1}, p_t^f, h_t^{enc}, y_t\right] + b_z^\phi\right), \tag{9}$$

by which we could calculate the parameters μ_t^ϕ, $\log(\sigma_t^\phi)^2$ and reparameterize the posterior of z_t by,

$$\begin{cases} \mu_t^\phi = W_{z,\mu}^\phi h_t^{z^\phi} + b_\mu^\phi \\ \log(\sigma_t^\phi)^2 = W_{z,\sigma}^\phi h_t^{z^\phi} + b_\sigma^\phi \end{cases} \Rightarrow z_t^{post} = \mu_t^\phi + \sigma_t^\phi \odot \epsilon, \tag{10}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is white Gaussian noise, $W_{z,\mu}^\phi, W_{z,\sigma}^\phi$ are weight matrices and $b_\mu^\phi, b_\sigma^\phi$ are bias vectors.

Similarly, for the prior, the hidden state of z_t is computed as $h_t^{z^\theta} = \tanh\left(W_z^\theta\left[z_{t-1}, p_t^f, h_t^{enc}\right] + b_z^\theta\right)$ and prior parameters μ_t^θ and $\log(\sigma_t^\theta)^2$ could be calculated by,

$$\begin{cases} \mu_t^\theta = W_{o,\mu}^\theta h_t^{z^\theta} + b_\mu^\theta \\ \log(\sigma_t^\theta)^2 = W_{o,\sigma}^\theta h_t^{z^\theta} + b_\sigma^\theta \end{cases} \Rightarrow z_t^{prior} = \mu_t^\theta + \sigma_t^\theta \odot \epsilon, \tag{11}$$

where $W_{o,\mu}^\theta, W_{o,\sigma}^\theta$ are weight matrices and $b_\mu^\theta, b_\sigma^\theta$ are bias vectors.

4.2.3 Recurrent Variational Decoder

To decode the movements from latent variables, we integrate a recurrent generative decoding strategy similar to the encoder. Formally, we need to model the conditional distribution $p_\theta\left(\boldsymbol{y} \mid \boldsymbol{P^f}, \boldsymbol{Z}\right)$.

For every time step before T, we decode the stock price movement prediction $\hat{y}_t$ by,

$$\begin{aligned} h_t^{dec} &= \tanh\left(W_{dec}\left[p_t^f, h_t^{enc}, z_t^{prior}, z_t^{post}\right] + b_{dec}\right), \\ \hat{y}_t &= \mathrm{softmax}\left(W_y h_t^{dec} + b_y\right), t < T, \end{aligned} \tag{12}$$

where W_{dec}, W_y are weight matrices and b_{dec}, b_y are bias vectors. The softmax function outputs the confidence distribution over up and down. As introduced before, we will leverage

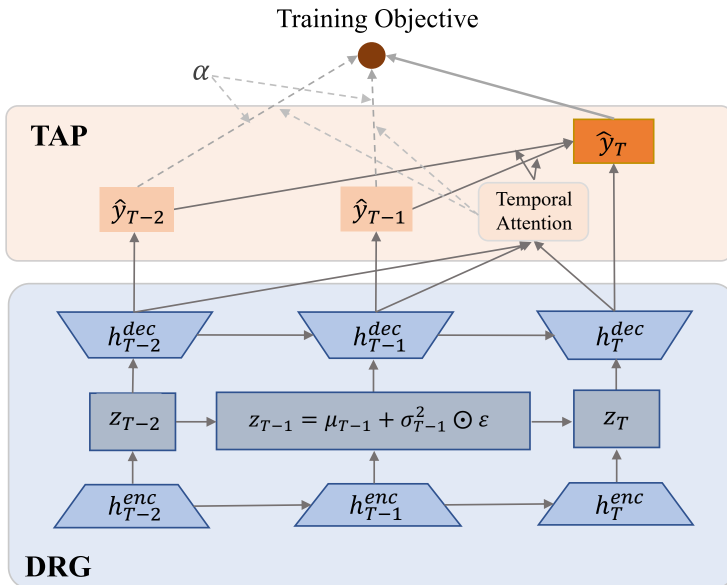

Figure 4: The architecture of the Deep Recurrent Generation (DRG) and the Temporal Attention Prediction (TAP) Modules.

the former auxiliary movement predictions $y_{<T} = \{y_t\}_{T-L+1}^{T-1}$ to enhance the performance of the target movement prediction y_T, this will be elaborated in the next section.

4.3 Temporal Attention Prediction

To explore the relationship between the main task, i.e. predicting target movement $\hat{y}_T$, with the auxiliary predictions, i.e. predicting the former movements $y_{<T} = \{y_t\}_{T-L+1}^{T-1}$, we adopt a temporal attention mechanism as shown in the red part of Figure 4.

Intuitively, each auxiliary prediction contributes unequally to the main prediction task for the different temporal dependencies. Thus, temporal attention weights are calculated by two evaluation aspects: the information score and the dependency score. The information score evaluates the information contributions from the lagged trading days to the target trading day, while the dependency score captures the dependency relations between them.

Specifically, the decoder hidden state $H^{dec} = [h_1^{dec}, \ldots, h_{T-1}^{dec}]$ is used to calculate the information score vector q^{dec}, dependency score vector k^{dec}, then the normalized attention weight vector w^{dec} could be calculated as follows,

$$\begin{aligned} q^{dec} &= (h_T^{dec})^\top \tanh\left(W_q H^{dec}\right), \\ k^{dec} &= w_k^\top \tanh\left(W_k H^{dec}\right), \\ w^{dec} &= \mathrm{softmax}\left(q^{dec} \odot (k^{dec})^\top\right), \\ v^{dec} &= [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{T-1}], \end{aligned} \tag{13}$$

where W_q, W_k are weight matrices, the weight vector w^{dec} measures the correlation between the target day and the past trading days.

Finally, the main prediction benefits from the auxiliary predictions, and we decode the final stock movement $\hat{y}_T$ as follows,

$$\hat{y}_T = \operatorname{softmax}(W_{dec}^T[v^{dec}(w^{dec})^\top, h_T^{dec}] + b_{dec}), \tag{14}$$

where W_{dec}^T is a weight matrix and b_{dec} is a bias vector.

4.4 Learning Objective

As introduced above, we leverage the variational approximator $q_\phi\left(\boldsymbol{Z} \mid \boldsymbol{P^f}, \boldsymbol{y}\right)$ to approximate the posterior distributions of latent variables $p_\theta\left(\boldsymbol{Z} \mid \boldsymbol{P^f}, \boldsymbol{y}\right)$. Then the likelihood of our target conditional probability distribution $p_\theta(\boldsymbol{y} \mid \boldsymbol{P^f})$ can be decomposed into:

$$\begin{aligned}
&\log p_\theta(\boldsymbol{y} \mid \boldsymbol{P^f}) \\
&= D_{\mathrm{KL}}\left[q_\phi(\boldsymbol{Z} \mid \boldsymbol{P^f}, \boldsymbol{y}) \| p_\theta(\boldsymbol{Z} \mid \boldsymbol{P^f}, \boldsymbol{y})\right] \\
&+ E_{q_\phi(\boldsymbol{Z}\mid\boldsymbol{P^f},\boldsymbol{y})}\left[\log p_\theta(\boldsymbol{y} \mid \boldsymbol{P^f}, \boldsymbol{Z})\right] \\
&- D_{\mathrm{KL}}\left[q_\phi(\boldsymbol{Z} \mid \boldsymbol{P^f}, \boldsymbol{y}) \| p_\theta(\boldsymbol{Z} \mid \boldsymbol{P^f})\right].
\end{aligned} \tag{15}$$

Therefore, maximizing the log-likelihood equals minimizing the Kullback-Leibler divergence, and also equals maximizing the last two terms of Equation 15. Thus, the variational recurrent lower bound could be decomposed as follows by plugging Equation 5 into Equation 15,

$$\begin{aligned}
&\mathcal{L}(\theta, \phi; \boldsymbol{P^f}, \boldsymbol{y}) \\
&= \sum_{t=T-L+1}^{T} E_{q_\phi\left(z_t \mid z_{<t}, p_{\leq t}^f, y_t\right)}\Bigg\{\log p_\theta\left(y_t \mid p_{\leq t}^f, z_*\right) \\
&\qquad\qquad - D_{\mathrm{KL}}\left[q_\phi\left(z_t \mid z_{<t}, p_{\leq t}^f, y_t\right) \big\| p_\theta\left(z_t \mid z_{<t}, p_{\leq t}^f\right)\right]\Bigg\} \\
&\leq \log p_\theta(\boldsymbol{y} \mid \boldsymbol{P^f}),
\end{aligned} \tag{16}$$

where the log-likelihood term is,

$$p_\theta\left(y_t \mid p_{\leq t}^f, z_*\right) = \begin{cases} p_\theta\left(y_t \mid p_{\leq t}^f, z_t\right), & \text{if } t < T. \\ p_\theta\left(y_T \mid p_{\leq t}^f, z_{\leq t}\right), & \text{if } t = T. \end{cases} \tag{17}$$

Compared to modeling the prior as a simple Standard Gaussian distribution, Equation 16 provides a more theoretically rigorous lower bound because of the prior $p_\theta\left(z_t \mid z_{<t}, p_{\leq t}^f\right)$ playing a dynamic role in inferring dependent latent variables for every different model input and latent history.

To incorporate both latent-space regularization and temporal importance into the learning objective, we use the per-time-step variational objective term. Following the KL annealing trick (Bowman et al., 2016; Semeniuta et al., 2017), we use a linearly increasing

KL weight $\lambda_{KL} \in (0, 1]$ to gradually release the KL regularization effect during training. Specifically, λ_{KL} starts from 0 at the beginning of training, increases linearly with a per-step rate of 0.005, and saturates at 1.0. This schedule prevents the KL term from dominating the objective too early and helps reduce the risk of posterior collapse.

With this annealed KL weight, the objective term at each time step is defined as:

$$f_t = \log p_\theta \left(y_t \mid p^f_{\leq t}, z_* \right) - \lambda_{KL} D_{\mathrm{KL}} \left[q_\phi \left(z_t \mid z_{<t}, p^f_{\leq t}, y_t \right) \| p_\theta \left(z_t \mid z_{<t}, p^f_{\leq t} \right) \right], \tag{18}$$

where the first term maximizes the prediction likelihood and the second term regularizes the variational posterior toward the conditional prior. Compared with using a fixed KL coefficient from the beginning of training, the annealing schedule allows the model to learn useful prediction signals at first and then progressively strengthen the latent-space regularization.

Next, to reflect the different importance of historical auxiliary predictions and the target prediction, we apply the temporal attention weights $w^{dec} \in \mathbb{R}^{1 \times T-1}$ to the objective terms. Specifically, we define the objective weight vector as

$$w^{obj} = [\lambda_{obj} w^{dec}, 1],$$

where $\lambda_{obj} \in (0, 1]$ controls the overall contribution of auxiliary prediction objectives, and the final value 1 represents the objective importance of the target prediction.

Finally, we could write the learning objective as follows,

$$\mathcal{L}(\theta, \phi; \boldsymbol{P^f}, \boldsymbol{y}) = \frac{1}{L} \sum_{t=T-L+1}^{T} w^{obj} f_t, \tag{19}$$

by which all the parameters $\{\theta, \phi\}$ could be updated through backpropagation. Besides, the whole process of the AgentPEN is summarized in Algorithm 2 in Appendix A for clarity.

5. Experiments

5.1 Experimental Setup

5.1.1 Datasets

To verify the effectiveness of the proposed method, we conducted experiments on three datasets. CMIN-US (Luo et al., 2023) dataset contains the top 110 stocks in the US market, with price data from Yahoo Finance and textual data from Yahoo and Wind. The time range of this dataset is from 2018-01-01 to 2021-12-31. CMIN-CN (Luo et al., 2023) contains the stock prices and text information of all 300 stocks in the CSI300 index in the Chinese market from 2018-01-01 to 2021-12-31. The stock price information comes from Yahoo and the text information from Wind[1].

Moreover, in this work, we construct a new dataset called AgentPEN-US by following the data collection process used for the ACL18 StockNet dataset (Xu and Cohen, 2018). The price data is collected from Yahoo Finance[2], while the news text data is collected

1. https://www.wind.com.cn/
2. https://finance.yahoo.com/

Table 1: Statistics of the three datasets. Up and Down columns report the label distribution (next-day movement).

Dataset	Split	Date range	Samples	Up	Down	Price Source	News Source
AgentPEN-US	Train	2014-01-01 – 2015-08-01	4,971	2,569	2,402	Yahoo Finance	Google News
	Dev	2015-08-01 – 2015-10-01	671	329	342		
	Test	2015-10-01 – 2015-12-31	984	562	422		
CMIN-US	Train	2018-01-01 – 2020-12-31	5,633	2,935	2,698	Yahoo Finance	Yahoo & Wind
	Dev	2021-01-01 – 2021-05-31	586	327	259		
	Test	2021-06-01 – 2021-12-31	862	460	402		
CMIN-CN	Train	2018-01-01 – 2020-12-31	3,506	1,732	1,774	Yahoo Finance	Yahoo & Wind
	Dev	2021-01-01 – 2021-05-31	821	387	434		
	Test	2021-06-01 – 2021-12-31	1,040	431	609		

from Google News[3]. This dataset includes 88 stocks, consisting of 8 stocks from the Conglomerates industry and the top 10 stocks by market capitalization from each of the other 8 industries: Basic Materials, Consumer Goods, Healthcare, Services, Utilities, Financial, Industrial Goods, and Technology. The detailed statistics and split protocol of the three datasets are shown in Table 1.

5.1.2 Baselines

To evaluate the effectiveness of the proposed AgentPEN, we selected 10 different baselines as follows:

- **ALSTM** (Qin et al., 2017) proposed a dual-stage attention-based recurrent neural network to select the most relevant stocks and used them to achieve long-term dependency modeling.
- **Adv-ALSTM** (Feng et al., 2019) added perturbations to simulate the stochasticity of price variables and trained the model under small yet intentional perturbations to tackle the overfitting problem.
- **HAN** (Hu et al., 2018) designed a Hybrid Attention Network to predict the stock trend based on recent related news.
- **StockNet** (Xu and Cohen, 2018) applied a Recurrently Variational Autoencoder to encode the price and news.
- **DTML** (Yoo et al., 2021) captured asymmetric and dynamic correlations between stocks using LSTM and Transformer.
- **PEN** (Li et al., 2023) used a shared representation learning module to characterize the interaction between text and price data.

3. https://news.google.com/

- **CMIN** (Luo et al., 2023) modeled the causality between financial texts and price features and designed a multi-memory network for multi-modal information interaction.
- **FinMA (PIXIU)** (Xie et al., 2023) fine-tuned LLaMA on FLARE, a large-scale multi-task instruction-tuning dataset, to improve its generalization ability across a wide range of financial tasks.
- **MagicNet** (Luo et al., 2025) considered both temporal and spatial dependencies in financial documents and introduced causality-based correlations between multivariate stocks hierarchically.
- **ExPred** (Li et al., 2026) combined a hybrid reflection agent with direct preference hierarchical optimization to train LLMs to generate explainable stock movement predictions.

5.1.3 Evaluation Metrics

We evaluate the accuracy of model results by two metrics: accuracy (ACC) and Matthews Correlation Coefficient (MCC) (Xu and Cohen, 2018) following the baselines.

Given the confusion matrix containing the numbers of true positive (tp), false positive (fp), true negative (tn), and false negative (fn) samples.

$$\begin{aligned} ACC &= \frac{tp+tn}{tp+tn+fp+fn}, \\ MCC &= \frac{tp \times tn - fp \times fn}{\sqrt{(tp+fp)(fn+tp)(fn+tn)(fp+tn)}} \end{aligned} \tag{20}$$

5.1.4 Implementation Details

AgentPEN is implemented using PyTorch with the Adam optimizer. The price feature dimension $d = 3$. We leverage grid search to select some important parameters. The look-back window L is selected from $[2, 3, 4, 5]$, and the best performance is achieved with $L = 5$. The batch size is set as 512. We chose a learning rate of 10^{-4} from $10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}$. The correlation threshold τ_c is set as 5 from $[1, 3, 5, 7, 9]$. The number of retrieved news items K is 10 selected from the range of $[5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$. The size of hidden states h_t^{enc} and the size of $h_t^{z^\phi}$ are set to 150, and the size of h_t^{dec} is set to 50. The λ_{obj} is 0.005. The hidden size of the Information Fusion Module d_{model} is set to 128. We use the LLM_{emb}=`gte-large` model [4] to encode the text into a 1024-dimensional embedding. For $LLM_{\text{selection}}$, LLM_{temporal} and LLM_{spatial}, we employ `gpt-3.5-turbo` as evaluators.

5.2 Results of Prediction Performance

Table 2 summarizes the experimental results on three different datasets using various baseline models and the proposed method, AgentPEN. From the results, we observe that AgentPEN consistently outperforms existing baselines across all three datasets in terms of both ACC and MCC. To further evaluate the robustness of AgentPEN, we additionally run the

4. https://huggingface.co/thenlper/gte-large

Model	AgentPEN-US		CMIN-US		CMIN-CN	
	ACC (%)	MCC	ACC (%)	MCC	ACC (%)	MCC
ALSTM	51.81	0.032	51.64	0.006	53.35	0.023
Adv-ALSTM	52.75	0.052	51.73	0.012	53.49	0.025
HAN	53.37	0.061	53.72	0.010	53.59	0.016
Stocknet	53.12	0.063	52.46	0.022	54.53	0.045
DTML	54.25	0.089	52.06	0.031	54.42	0.083
PEN	53.95	0.071	53.20	0.027	54.83	0.086
CMIN	54.65	0.085	53.43	0.046	55.28	0.111
FinMA	49.13	0.008	48.87	-0.057	47.03	0.003
MagicNet	54.79	0.087	54.22	0.049	55.53	0.126
ExPred	55.14	0.109	55.85	0.121	55.67	0.089
AgentPEN	**56.91**	**0.118**	**58.09**	**0.130**	**57.09**	**0.128**

Table 2: Performance comparison of AgentPEN and baselines.

model with five different random seeds and report the corresponding distributions in Figure 5. The left part of the figure shows the ACC distributions, while the right part presents the MCC distributions. AgentPEN achieves average ACC values of 56.91 ± 0.71, 58.09 ± 0.64, and 57.09 ± 0.28 on AgentPEN-US, CMIN-US, and CMIN-CN, respectively, and average MCC values of 0.1178 ± 0.0137, 0.1296 ± 0.0157, and 0.1282 ± 0.0045. These results demonstrate that AgentPEN maintains stable performance across different random initializations. Moreover, compared with two strong baselines, MagicNet and ExPred, whose performances are shown as horizontal dashed lines in Figure 5, AgentPEN achieves consistently better results on both ACC and MCC across all datasets. Overall, the consistent advantages of AgentPEN across different markets, metrics, and random seeds suggest its effectiveness and robustness as a stock movement prediction model.

5.3 Ablation Study

To verify the effectiveness of the proposed method, we compared the model's performance by removing each component.

- **AgentPEN-TSM**: AgentPEN without the Text Selection Module (TSM), i.e., disabling the selection module by setting $\tau_c = 0$, so that all retrieved news items are forwarded without filtering.
- **AgentPEN-random**: replaces each correlation score with a random score sampled independently from $\mathcal{U}(0, 10)$ and then applies the same threshold $\tau_c = 5$.
- **AgentPEN-TMM**: AgentPEN without the Text Memory Module (TMM), i.e. we only leverage news in the look-back window to serve market information.
- **AgentPEN-IFM**: AgentPEN without the Information Fusion Module (IFM), i.e. the price embedding and the news embedding are simply concatenated to form the final market information representation.

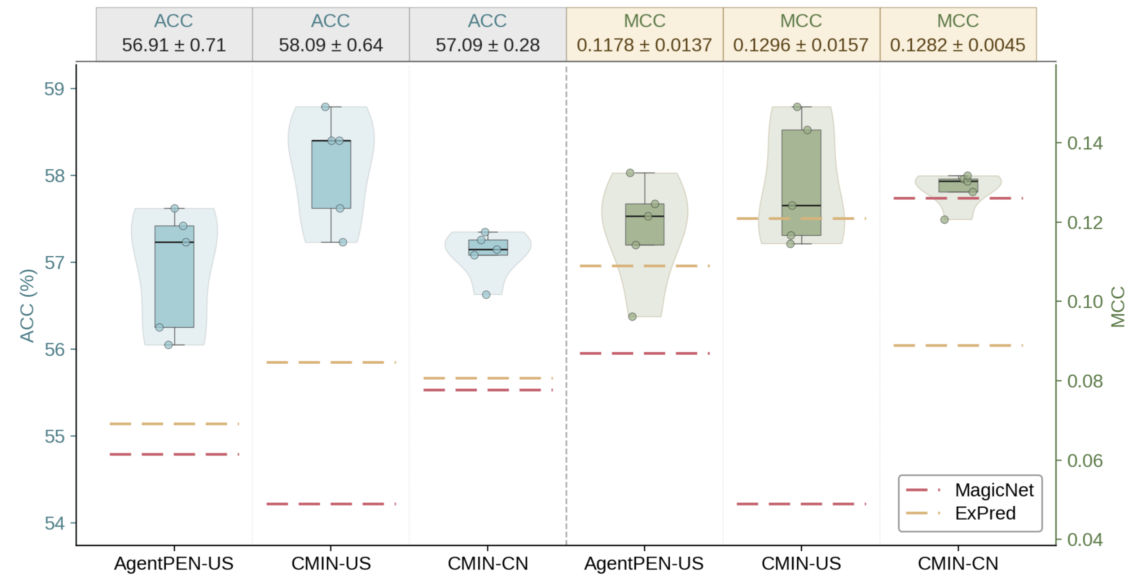

Figure 5: Performance distributions of AgentPEN over five random seeds on three datasets. The left panel reports ACC, and the right panel reports MCC. Horizontal dashed lines indicate the performance of two strong baselines, MagicNet and ExPred.

- **AgentPEN-price**: We only leverage the price data as inputs and neglect the news corpora.
- **AgentPEN-news**: AgentPEN without price data, and obtains results solely based on news text information.
- **AgentPEN-Deterministic**: AgentPEN with a deterministic variant of the Deep Recurrent Generation module. The prior and posterior networks are kept, but the stochastic latent variable is replaced by its mean, i.e., $z_t := \mu_t^\phi$ during training and $z_t := \mu_t^\theta$ at test time, and the KL term is removed by setting $\lambda_t \equiv 0$.
- **AgentPEN-MLP**: AgentPEN without the VAE-based latent variable in the Deep Recurrent Generation module. The fused market-information sequence $\{p_t^f\}$ is passed through the same GRU and then a two-layer MLP classifier, without prior/posterior networks, stochastic sampling, or KL regularization.

The ablation study results are shown in Table 3. Overall, removing or weakening any key component leads to consistent performance degradation, demonstrating that each module contributes to AgentPEN's final prediction performance. First, the comparison between AgentPEN, AgentPEN-TSM, and AgentPEN-Random verifies the effectiveness of the LLM-based Text Selection Module. AgentPEN-TSM disables news filtering and forwards all

Model	AgentPEN-US		CMIN-US		CMIN-CN	
	ACC (%)	MCC	ACC (%)	MCC	ACC (%)	MCC
AgentPEN-TSM	54.10	0.073	53.71	0.091	51.95	0.043
AgentPEN-Random	54.13	0.074	53.61	0.083	51.75	0.041
AgentPEN-TMM	53.91	0.076	52.54	0.083	51.66	0.036
AgentPEN-IFM	53.32	0.067	52.34	0.030	51.86	0.040
AgentPEN-price	53.22	0.069	53.12	0.077	51.66	0.038
AgentPEN-news	52.64	0.053	52.73	0.045	51.37	0.031
AgentPEN-Deterministic	55.31	0.070	55.00	0.076	50.78	0.048
AgentPEN-MLP	55.35	0.009	56.68	0.010	51.77	0.051
AgentPEN	**56.91**	**0.118**	**58.09**	**0.130**	**57.09**	**0.128**

Table 3: Ablation study results.

retrieved news, while AgentPEN-Random keeps approximately the same filtering ratio as AgentPEN but replaces the LLM-produced correlation scores with random scores. Both variants perform substantially worse than the full model across all datasets. For example, on CMIN-US, AgentPEN improves ACC from 53.71% and 53.61% to 58.09%, and MCC from 0.091 and 0.083 to 0.130, compared with AgentPEN-TSM and AgentPEN-Random, respectively. This indicates that the improvement does not simply come from reducing the number of news items, but from the relevance information provided by the LLM-produced correlation scores.

Second, removing the Text Memory Module also degrades performance, showing that temporal-spatial memory helps retrieve useful news beyond naive look-back-window modeling and captures broader market influence. Third, when the Information Fusion Module is removed, the model replaces cross-attention with simple concatenation followed by a linear layer, leading to clear performance drops on all datasets. This confirms that explicitly modeling text-price interactions is more effective than simple feature concatenation. Fourth, the single-modality variants, AgentPEN-price and AgentPEN-news, perform worse than the full model, indicating that both textual news and numerical price information are important. In particular, the performance decline of AgentPEN-news highlights the central role of price features, while the gap between AgentPEN-price and the full model shows that textual news provides complementary information to price sequences.

Finally, we further examine the contribution of the VAE-based Deep Recurrent Generation module through AgentPEN-Deterministic and AgentPEN-MLP. AgentPEN-Deterministic keeps the prior and posterior networks but replaces the stochastic latent variable with its mean and removes the KL term. This variant consistently underperforms the full model on all datasets, reducing MCC from 0.118, 0.130, and 0.128 to 0.070, 0.076, and 0.048 on AgentPEN-US, CMIN-US, and CMIN-CN, respectively. This suggests that stochastic latent modeling is useful for capturing uncertainty in stock movements. AgentPEN-MLP removes the latent variable entirely and uses a GRU followed by a two-layer MLP classifier. Although this variant achieves relatively competitive ACC on some datasets, its MCC drops sharply to 0.009, 0.010, and 0.051, indicating that a simple deterministic discriminative

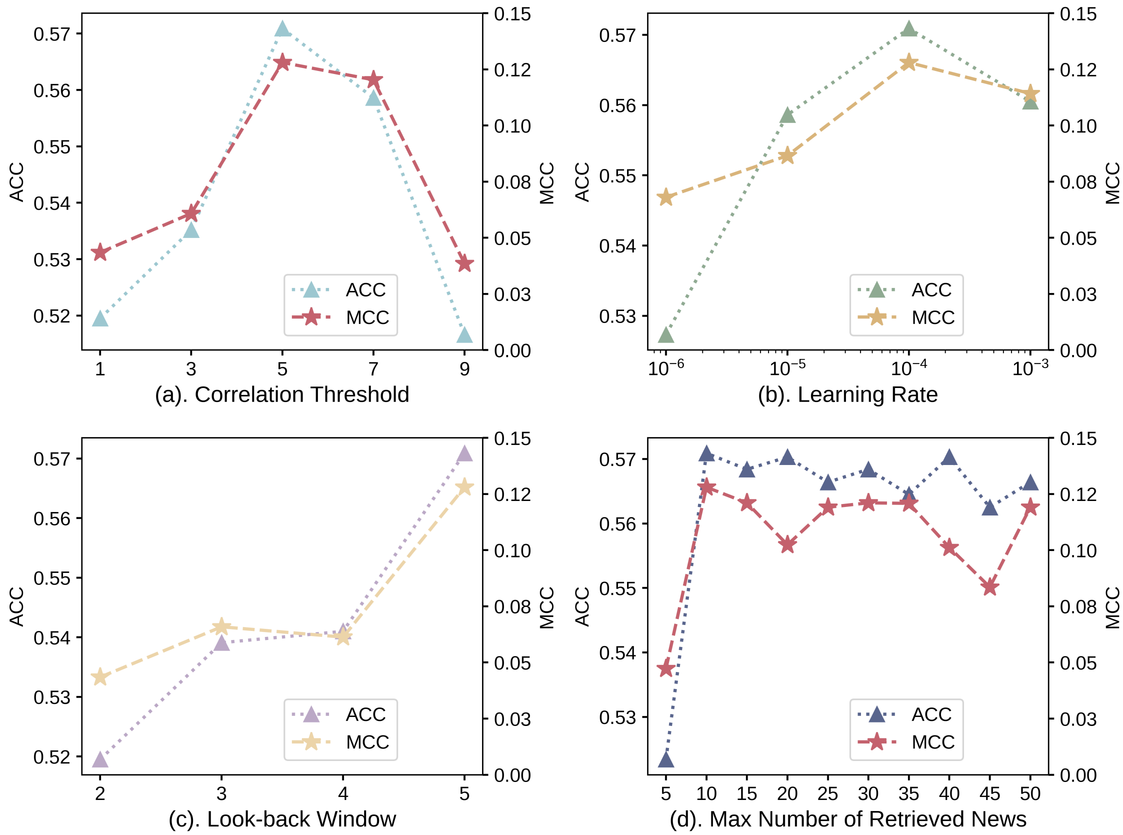

Figure 6: The parameter sensitivity study for four important parameters: (a) Correlation threshold τ_c in Text Selection Module; (b) Learning rate; (c) Look-back window; (d) Max number of retrieved news.

head is insufficient for balanced stock movement prediction. These results show that the conditional VAE is not merely an architectural complication; rather, its stochastic latent variables help improve balanced classification and prevent the model from degenerating into weak or biased decision behavior. Collectively, these ablation results demonstrate the necessity of LLM-based selection, temporal-spatial memory, cross-modal fusion, multimodal input integration, and VAE-based stochastic movement modeling in AgentPEN.

5.4 Parameter Sensitivity Study

We conduct a parameter sensitivity study on dataset CMIN-CN to examine how the correlation threshold, learning rate, look-back window size and the max number of retrieved news affect the performance of AgentPEN. The study results are shown in Figure 6.

The correlation threshold τ_c is a parameter used to filter news articles based on their correlation with the target stock. At lower thresholds, the model may include too much irrelevant information, which dilutes the meaningful signals. Conversely, at higher thresholds, the model might discard too much information, leading to a loss of valuable signals. Therefore, a correlation threshold of 5 was selected based on Figure 6(a). Figure 6(b) shows the performance of AgentPEN across different learning rates. The best performance is obtained at 10^{-4}. The look-back window represents the number of past days considered when inputting information into the model. As shown in Figure 6(c), both ACC and MCC metrics initially increase as the look-back window expands from 2 to 5. We choose a look-back window of 5, which strikes an optimal balance between leveraging sufficient historical data to capture meaningful trends and avoiding the inclusion of outdated information. The maximum number of retrieved news articles controls how many pieces of news are pulled from memory to inform the model's predictions. A larger number provides more information but also increases the risk of introducing noise and computational complexity. Based on Figure 6(d), 10 was selected as the optimal parameter considering both the performance and the computational efficiency.

5.5 Results of Explainability

We evaluate the explainability of AgentPEN from three perspectives using case studies from the CMIN-US dataset: the explainability of selected texts, the explainability of Temporal & Spatial Memory, and the explainability of retrieved news. In these cases, the target is the stock movement of $AAPL on the trading date 2018/02/23. The look-back window size is set as 3, i.e., the historical time range is from 2018/02/20 to 2018/02/22 for clarity.

5.5.1 The Explainability of Text Selection Module

The Text Selection Module filters the incoming news data and selects news items with high correlation to the target stock. As analyzed before, the correlation threshold is chosen as $\tau_c = 5$. To quantitatively evaluate whether the LLM-produced correlation score reflects meaningful market relevance, we conduct a population-level correlation-bucket analysis. For each test sample, we take the maximum LLM-assigned correlation score among all look-back news items and group samples into five buckets: low (0), low (1–3), mid (4–6), high (7–8), and very-high (9–10). For each bucket, we compute the mean absolute next-day return. The intuition is that if the LLM relevance score captures meaningful market-related information, then samples containing highly relevant news should be associated with stronger subsequent price reactions.

As shown in Figure 7, higher LLM-assigned correlation buckets are generally associated with larger next-day price movements across the three datasets. On AgentPEN-US, the mean absolute next-day return increases from 1.27% in the low (0) bucket to 1.50% in the very-high (9–10) bucket. On CMIN-US, it increases from 1.23% in the low-correlation bucket to 1.87% in the very-high bucket. On CMIN-CN, the trend is also clear: the mean absolute return increases from 2.17% in the mid (4–6) bucket to 2.44% in the very-high bucket. These results indicate that the LLM-produced relevance signal is empirically associated with the magnitude of subsequent market reactions, rather than being an arbitrary textual label. This population-level analysis provides quantitative evidence for the explanatory relevance of the Text Selection Module and complements the qualitative case study below.

Table 4: The case study of explainability for the process of the text selection evaluation and the temporal-spatial evaluation. We present a sample of news data for illustration. The target stock and target date are $AAPL and 2018/02/23, respectively. The look-back window size is set as 3, i.e., the look-back window is from 2018/02/20 to 2018/02/22.

Date	News	Text Selection Evaluation		Temporal Memory Evaluation		Spatial Memory Evaluation
		Correlation	**Selected**	**Duration**	**Impact Range**	**Affected Stock**
2018/02/20	Apple Loop: Apple Confirms Serious iPhone X Problem, MacBook Pro's Mistakes, iOS Development Slows.	10	✓	25	2018/02/20 - 2018/03/17	$AAPL
2018/02/20	Warren Buffett takes a big bite of Apple, dumps IBM (Video).	10	✓	15	2018/02/20 - 2018/03/07	$AAPL / $IBM
2018/02/20	A former Google and Apple exec now teaches Stanford students to design products to make people happy.	3	✗	-	-	-
2018/02/20	Does the 6.1-inch LCD iPhone Replace iPhone 8 or iPhone 8 Plus?	10	✓	8	2018/02/20 - 2018/02/28	$AAPL
2018/02/20	Apple Leak Reveals Significant iPhone Design Changes.	10	✓	28	2018/02/20 - 2018/03/20	$AAPL
2018/02/21	Apple Inc. Stock Looks Cheap, but It Still Has Risks.	10	✓	21	2018/02/21 - 2018/03/14	$AAPL
2018/02/21	Spotify Is About to Start Making Hardware.	3	✗	-	-	-
2018/02/21	Alphabet Shares Rise on Google Pay Debut: Should Apple Be Worried?	6	✓	18	2018/02/21 - 2018/03/11	$AAPL
2018/02/21	Apple Is Officially Taking Over the Watch Business.	10	✓	23	2018/02/21 - 2018/03/16	$AAPL
2018/02/21	It's time to update your iPhone - Apple just fixed a big text message bug.	9	✓	26	2018/02/21 - 2018/03/19	$AAPL
2018/02/22	Wireless Carriers Sold a Third of All Apple Watches in the U.S. Last Quarter.	8	✓	15	2018/02/22 - 2018/03/09	$AAPL / $T / $VZ
2018/02/22	Apple in Talks With Cobalt Mines to Secure Supply Chain.	10	✓	30	2018/02/22 - 2018/03/24	$AAPL / $CAT / $BHP / $RIO
2018/02/22	Facebook, Alphabet defy stock market dip as Apple, other Silicon Valley stocks fall.	10	✓	21	2018/02/22 - 2018/03/15	$AAPL / $FB
2018/02/22	Apple OLED-Screen iPhone Production Plans Shrinking.	10	✓	10	2018/02/22 - 2018/03/04	$AAPL
2018/02/22	Why AT&T's 5G Network Won't Be Speeding Up Your Phone Anytime Soon.	3	✗	-	-	-

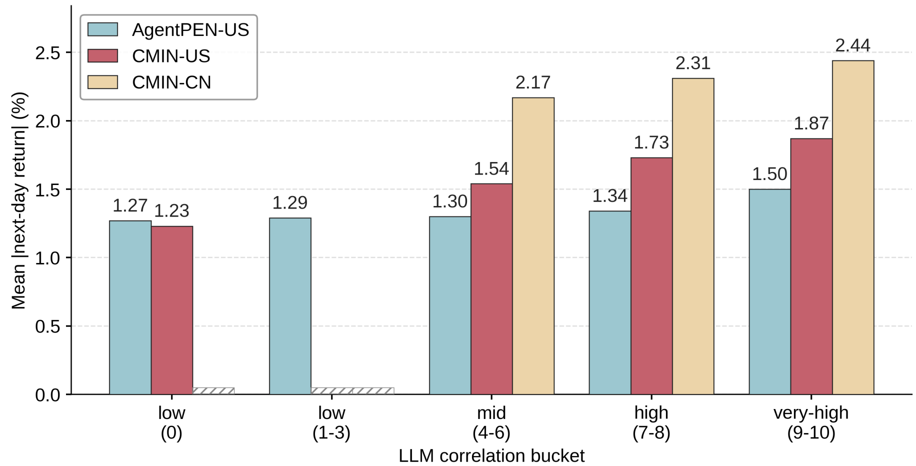

Figure 7: Quantitative evaluation of the Text Selection Module. Test samples are grouped by the maximum correlation score among their look-back news, and the y-axis reports the mean absolute next-day return. Samples with higher correlation scores generally show larger subsequent price movements, indicating that the LLM-based selection mechanism captures prediction-relevant news signals.

For qualitative illustration, Table 4 presents an example for the target stock $AAPL. Out of the 15 news items listed, 12 are selected according to their correlation scores. For example, news such as "Apple Confirms Serious iPhone X Problem" and "Apple Leak Reveals Significant iPhone Design Changes" receive the maximum correlation score and are selected for further processing. In contrast, "A former Google and Apple exec now teaches Stanford students to design products to make people happy" is filtered out due to its weak relevance to the stock trend. This example shows how the Text Selection Module filters out less relevant news and retains news items with higher potential to support the prediction of $AAPL's stock movement.

5.5.2 The Explainability of Temporal-Spatial Memory Module

The Temporal Memory Module evaluates the duration of the impact that selected news items might have on the stock price over time. It could incorporate useful information that lies outside the look-back window. Table 4 shows the evaluation results of each news. For example, the news item "Apple Loop: Apple Confirms Serious iPhone X Problem, MacBook Pro's Mistakes, iOS Development Slows" has a long-lasting effect on $AAPL, while the impact duration of "It's time to update your iPhone - Apple just fixed a big text message bug." is shorter. The Temporal Memory Module allows the model to account for the sustained influence of significant news over time, ensuring that these effects are considered in the prediction process, which is crucial for accurate forecasting.

The Spatial Memory Module evaluates the broader market impact of the news, identifying other stocks that may be affected by the same news item. From Table 4, for instance, the news item "Warren Buffett takes a big bite of Apple, dumps IBM" collected for $AAPL also has a strong effect on stock $IBM, reflecting the interconnectedness of the market. Other examples include news that affects multiple stocks, such as "Apple in Talks With Cobalt Mines to Secure Supply Chain." may affect $AAPL / $CAT / $BHP / $RIO, showing how news can have widespread impacts across different sectors. Apple's stock may benefit from enhanced supply chain stability and reduced risk, while Caterpillar could see increased demand for mining equipment. BHP Billiton and Rio Tinto could gain from potential new contracts and increased market demand for cobalt, boosting their stock prices. The Spatial Memory Module enhances the model's understanding of how news from one stock can also influence other stocks, which is essential for a comprehensive market analysis. This broader view helps the model to factor in potential cross-stock and cross-sector impacts, making predictions more robust.

5.5.3 The Explainability of Retrieved News

To analyze the effectiveness of the Temporal-Spatial Memory retrieval process within the LLM-based Representation Fusion Agent (RFA), we present the news retrieved from Temporal Memory (retrieving news of $AAPL from dates outside the look-back window) and Spatial Memory (retrieving news of other stocks that could also affect $AAPL) in Table 5.

The top half of Table 5 presents some news texts retrieved from dates before the look-back window but still have an impact on $AAPL. For example, the news item "How the New Tax Code Will Affect Tech Giants' Earnings." was created on 2018/02/06, which is out of the look-back window range but has a lasting impact on $AAPL. If the new tax code includes lower corporate tax rates or other tax benefits for large tech companies like Apple, this could lead to an increase in Apple's net earnings. Higher earnings typically result in a positive sentiment among investors, potentially leading to an increase in the stock price.

The bottom half of Table 5 presents some texts belonging to other stocks that might influence $AAPL. For instance, the news item "Apple, Amazon stocks bounce sharply to swing into positive territory." belongs to $AMZN while related to $AAPL. This news presents a strong potential signal indicating a significant upward trend in Apple's stock price.

The retrieval from Temporal and Spatial Memory is effective in broadening the model's awareness of market dynamics. Under such a dual-level memory mechanism, the model can better anticipate indirect effects, ensuring a more comprehensive analysis of factors influencing the stock.

5.6 Investment Simulation

Beyond ACC and MCC, we further conduct an investment simulation to evaluate whether AgentPEN's binary movement predictions can be translated into trading-oriented performance. We focus on the CMIN-CN test window from 2021-06 to 2021-12, since this period lies in a more challenging non-trending and bearish regime. Therefore, CMIN-CN provides a more informative setting for examining whether AgentPEN can provide useful directional signals under adverse market conditions.

For each trading day, we translate the predicted up/down movements into daily-rebalanced equally weighted portfolios. We consider three strategies. The AgentPEN long-short strategy takes long positions on stocks predicted to rise and short positions on stocks predicted

Table 5: The case study of explainability for the retrieval process of target stock $AAPL and date 2018/02/23. News texts are retrieved from two aspects: from other dates outside the look-back window (based on Temporal Memory) and from other stocks (based on Spatial Memory). The look-back window is set from 2018/02/20 to 2018/02/22.

	Stock	**Date**	**News**	**Duration**	**Impact Range**	**Affected Stock**
From Temporal Memory	$AAPL	2018/02/05	Apple Sets Records With Its Best iPhone Ever.	19	2018/02/05 - 2018/02/24	$AAPL
	$AAPL	2018/02/06	How the New Tax Code Will Affect Tech Giants' Earnings.	30	2018/02/06 - 2018/03/08	$AAPL / $AMZN / $MSFT / $FB
	$AAPL	2018/02/06	What's Next for Apple after Paying a $38 Billion Tax Bill?	28	2018/02/06 - 2018/03/06	$AAPL
	$AAPL	2018/02/07	Apple's Balance Sheet To Remain 'Robust,' Says Moody's.	17	2018/02/07 - 2018/02/24	$AAPL
From Spatial Memory	$AMZN	2018/02/06	Apple, Amazon stocks bounce to swing into positive territory.	25	2018/02/06 - 2018/03/03	$AMZN / $AAPL
	$BA	2018/02/01	Stocks Rise As Boeing, Apple Boost Dow; Bitcoin Play Sinks.	28	2018/02/01 - 2018/03/01	$BA / $AAPL
	$FB	2018/02/21	Is Facebook's Stock Now The Next Apple?	30	2018/02/21 - 2018/03/23	$FB / $AAPL
	$MSFT	2018/02/14	The Zacks Analyst Blog Highlights: Microsoft, Apple, Micron Technology, Lam Research and ASML Holding.	25	2018/02/14 - 2018/03/11	$MSFT / $AAPL

Strategy	Cum. return	Annual. Sharpe	Max DD	Win rate
Buy-and-hold	-5.77%	-2.77	-9.6%	38.5%
AgentPEN long-only	$-3.66 \pm 0.99\%$	-1.58 ± 0.49	-7.7%	46.2%
AgentPEN long-short	$\mathbf{+2.13 \pm 0.61\%}$	$\mathbf{2.47 \pm 0.54}$	$\mathbf{-2.2\%}$	**61.5%**

Table 6: Investment results of AgentPEN on CMIN-CN from 2021-06 to 2021-12. AgentPEN-based strategies are averaged over 5 seeds. The long-short strategy takes long positions on stocks predicted to rise and short positions on stocks predicted to fall, while the long-only strategy only holds stocks predicted to rise.

to fall. The AgentPEN long-only strategy only holds stocks predicted to rise and does not short stocks predicted to fall. The Buy-and-hold benchmark equally holds all stocks in the CMIN-CN stock universe throughout the test period without using model predictions. We report cumulative return, annualized Sharpe ratio, maximum drawdown (Max DD), and win rate.

Let r_t^p denote the portfolio return on trading day t, and let N be the number of trading days in the test period. The cumulative return is computed as: $R_{\text{cum}} = \prod_{t=1}^{N}(1 + r_t^p) - 1$.

The annualized Sharpe ratio is computed as: Sharpe $= \frac{\bar{r}^p}{\sigma(r^p)}\sqrt{252}$, where $\bar{r}^p$ and $\sigma(r^p)$ denote the mean and standard deviation of daily portfolio returns, respectively, and 252 is the commonly used number of trading days in one year. The maximum drawdown is computed as: $\text{MaxDD} = \min_{1 \le t \le N}\left(\frac{V_t}{\max_{1 \le s \le t} V_s} - 1\right)$, where $V_t = \prod_{i=1}^{t}(1 + r_i^p)$ denotes the portfolio value at day t. The win rate is computed as: $\text{WinRate} = \frac{1}{N}\sum_{t=1}^{N}\mathbb{I}\left(r_t^p > 0\right)$, where $\mathbb{I}(\cdot)$ is the indicator function.

As shown in Table 6, the CMIN-CN test period is unfavorable for passive holding: buy-and-hold obtains a negative cumulative return of -5.77%, a negative Sharpe ratio of -2.77, and a maximum drawdown of -9.6%. In this bearish regime, AgentPEN's long-short strategy is the only strategy that achieves a positive return across the tested seeds, with a cumulative return of $+2.13 \pm 0.61\%$, an annualized Sharpe ratio of 2.47 ± 0.54, a shallow maximum drawdown of -2.2%, and a win rate of 61.5%. This indicates that AgentPEN's predictions contain useful directional information that can be exploited by a market-neutral portfolio. The AgentPEN long-only strategy also reduces the loss compared with buy-and-hold, improving cumulative return from -5.77% to $-3.66 \pm 0.99\%$, and reducing maximum drawdown from -9.6% to -7.7%.

Figure 8 further visualizes the cumulative returns of the three strategies during the CMIN-CN test window. We shade four COVID-related sub-periods, including the reopening period, the Delta-surge period, the mid-recovery period, and the Omicron period. The performance difference across these periods suggests that AgentPEN's trading value is regime-dependent. During the reopening and mid-recovery periods, market conditions are relatively more stable, and the advantage of AgentPEN long-short over buy-and-hold is moderate. In such periods, broad market exposure is less harmful, so passive holding and long-only positioning can still benefit from partial recovery momentum.

In contrast, during the Delta-surge and Omicron periods, market uncertainty becomes stronger, and passive exposure to the whole market becomes riskier. This is reflected by

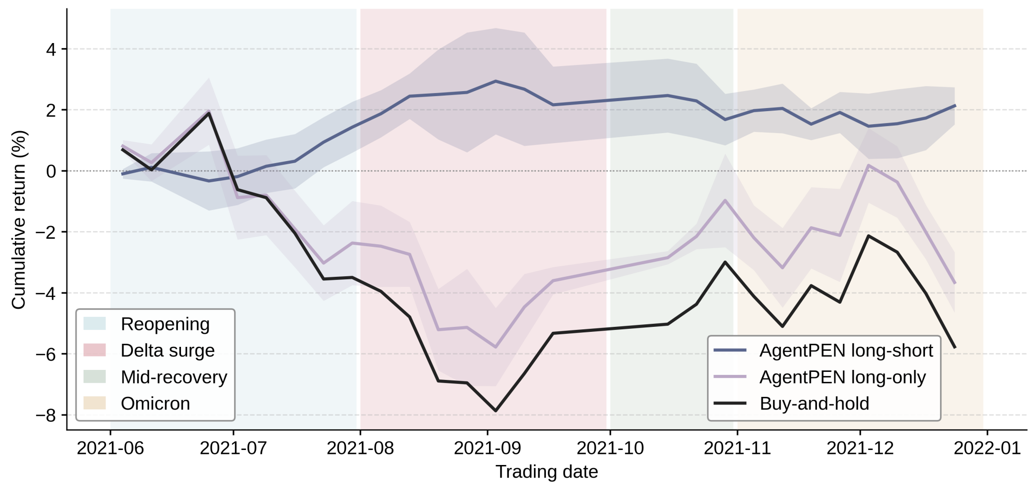

Figure 8: Cumulative returns of different investment strategies on CMIN-CN during the 2021-06 to 2021-12 test window. The shaded regions indicate four COVID-related market sub-periods.

the continued weakness of buy-and-hold and the negative performance of the long-only strategy. AgentPEN long-short performs better in these periods because it can use the model's directional predictions on both sides: taking long positions in stocks predicted to rise and short positions in stocks predicted to fall. Therefore, it can reduce market-wide downside exposure and exploit cross-sectional differences among stocks. The larger gap between AgentPEN long-short and the other strategies during the Delta-surge and Omicron periods suggests that AgentPEN's prediction signals are most useful under volatile or bearish market regimes, where identifying relative winners and losers is more valuable than simply holding the market.

5.7 Computational Cost

Since AgentPEN leverages LLMs to obtain structured textual signals from financial news, we further analyze its computational cost and scalability. In our implementation, the three LLM-based annotations, i.e., news relevance, impact duration, and affected stocks, are obtained through a single combined LLM call. Specifically, each retained news item is sent to `gpt-3.5-turbo` once, and the model returns a JSON object containing all three fields. This design reduces the number of LLM API calls by approximately $3\times$ compared with a naive implementation that separately queries $LLM_{\text{selection}}$, LLM_{temporal}, and LLM_{spatial}. The exact prompt is provided in Table 8 in Appendix B.

We profile the LLM cost on the three released datasets and report the statistics in Table 7. In total, AgentPEN processes 519,291 news items, corresponding to 268.3M input tokens and 31.1M output tokens. Based on the public `gpt-3.5-turbo` price of \$0.50/M input tokens and \$1.50/M output tokens at the time of our experiments, the total estimated annotation cost is \$180.89. The dataset-level costs are \$48.25 for AgentPEN-US, \$66.42 for

CMIN-US, and \$66.22 for CMIN-CN. The text embedding model `gte-large` is run locally and thus does not incur API cost.

Dataset	Samples	News	Avg. token per-news	Input token	Output token	`gpt-3.5-turbo`
AgentPEN-US	6,626	137,377	22	71.8M	8.2M	**\$48.25**
CMIN-US	7,081	190,847	16	98.5M	11.5M	**\$66.42**
CMIN-CN	5,367	191,067	13	98.0M	11.5M	**\$66.22**
Total	**19,074**	**519,291**	—	**268.3M**	**31.1M**	**\$180.89**

Table 7: LLM-call statistics and estimated API cost for LLM-based agent.

It is important to distinguish historical evaluation from real-time deployment. For historical evaluation and repeated backtesting, LLM annotations and text embeddings can be pre-computed and cached, so the same historical news items do not need to be repeatedly sent to the LLM. However, in a real deployment scenario, newly arrived daily news still needs to be annotated once before prediction, so AgentPEN's practical inference pipeline consists of two stages: an LLM-based update stage for new news and a neural forward stage for stock movement prediction. After the LLM-based news update is completed, the neural prediction component is efficient. On a single NVIDIA A100 80GB GPU, the forward time for a batch of 512 samples is 932ms, corresponding to 1.82ms per prediction and approximately 549 predictions per second.

The main scalability bottleneck of AgentPEN therefore lies in the LLM-based annotation of newly arrived news. Several strategies can mitigate this cost in deployment. First, news-level caching allows each news item's structured annotation to be reused across retraining, backtesting, and downstream tasks. Second, the combined-field prompt extracts all three structured signals in one call, reducing the number of API calls compared with separate prompts. Third, daily news can be annotated in batches and updated periodically. Fourth, smaller open-source or domain-specific financial LLMs can be used for structured extraction, since the key requirement is stable JSON-style annotation rather than open-ended generation. Finally, for latency-sensitive scenarios, a lightweight student model can be trained to approximate LLM-produced annotations, while only uncertain or high-impact news is routed to a larger LLM.

Overall, AgentPEN involves a practical trade-off: LLM calls are still required for newly arriving news during online deployment, but these calls can be reduced and amortized through combined prompting, caching, batching, and model distillation.

6. Conclusion

This study has introduced AgentPEN, a novel prediction-explanation network designed to enhance the accuracy and explainability of stock price movement predictions. By designing an LLM-based Representation Fusion Agent in a Selection-Memory-Fusion manner, AgentPEN effectively selects and integrates relevant information from vast and variable text sources, aligning them with stock price data through a cross-attention mechanism. In addition, we have released a new dataset, called AgentPEN-US, for further Fintech research. The model's superior performance in both prediction accuracy and explainability, as demonstrated across multiple real-world datasets, underscores its potential as a useful tool for financial institutions seeking to make informed investment decisions while satisfying audit and regulatory demands for transparency.

Appendix A. Algorithms

Algorithm 1 The detailed algorithm of LLM-based Representation Fusion Agent

Input: News text set $\boldsymbol{C} = \{c_t\}_{t=T-L+1}^{T}$ with $c_t = [c_t^1, \ldots, c_t^M]$; price sequence $\boldsymbol{P} = \{p_t\}_{t=T-L+1}^{T}$; target stock s; target trading day T; correlation threshold τ_c; retrieved news number K; $LLM_{\text{selection}}$, LLM_{temporal}, LLM_{spatial}, and LLM_{emb}.

Output: Fused representation $\boldsymbol{P}^f \in \mathbb{R}^{L \times d_{\text{model}}}$.

```
Initialize selected news set C^s ← ∅, temporal memory M_temp ← ∅, and spatial memory M_spat ← ∅.
// Text Selection
for t = T − L + 1 to T do
    for m = 1 to M do
        ρ_t^m ← LLM_selection(c_t^m, s).
        if ρ_t^m ≥ τ_c then
            Add c_t^m to C^s.
        end if
    end for
end for
// Temporal-Spatial Memory Writing
for each selected news item c_t^i ∈ C^s do
    d_t^i ← LLM_temporal(c_t^i, s),    r_t^i ← LLM_spatial(c_t^i, s).
    M_temp ← M_temp ∪ {(c_t^i, t, t + d_t^i, ρ_t^i)}.
    M_spat ← M_spat ∪ {(c_t^i, r_t^i, ρ_t^i)}.
end for
// Memory Retrieval
Build candidate set from Temporal Memory and Spatial Memory:
```

$$\mathcal{A}_{s,T} = \left\{ c_t^i \;\middle|\; (c_t^i, t, t + d_t^i, \rho_t^i) \in \mathcal{M}_{\text{temp}}, (c_t^i, r_t^i, \rho_t^i) \in \mathcal{M}_{\text{spat}}, T \in [t, t + d_t^i], s \in r_t^i \right\}.$$

```
Retrieve top-K candidates ranked by correlation score: C^k = TopK_{c_t^i ∈ A_{s,T}}(ρ_t^i).
If |C^k| < K, pad C^k with empty placeholders.
// Information Fusion
Encode and project retrieved news:
```

$$\boldsymbol{C}^l = LLM_{\text{emb}}(\boldsymbol{C}^k), \qquad \boldsymbol{C}^e = \boldsymbol{C}^l W_t + b_t.$$

```
Project price features:
```

$$\boldsymbol{P}^e = \boldsymbol{P} W_p + b_p.$$

```
Obtain fused representations by cross-attention:
```

$$\boldsymbol{P}^f = \text{MultiheadAtt}(\boldsymbol{P}^e, \boldsymbol{C}^e),$$

```
where P^e serves as the query and C^e serves as the key and value.
return P^f.
```

Algorithm 2 The whole process of AgentPEN.

Input: Stock price data $\boldsymbol{P} = \{p_t\}_{t=T-L+1}^{T} \in \mathbb{R}^{D \times L \times d}$ and news text set $\boldsymbol{C} = \{c_t\}_{t=T-L+1}^{T} \in \mathbb{R}^{D \times L \times M}$ with D stocks, look-back window L, stock price feature dimension d, and number of news items M in each day.

Output: The prediction of stock movement y_T.

1: // **LLM-based Representation Fusion Agent (Section 4.1)**

2: Obtain the fused market representation $\boldsymbol{P}^f$ by applying the LLM-based Representation Fusion Agent to the news text set $\boldsymbol{C}$ and price sequence $\boldsymbol{P}$, as detailed in Algorithm 1.

3: // **Deep Recurrent Generation (Section 4.2)**

4: For each time step t, calculate the hidden state h_t^{enc} of the encoder GRU and the hidden state $h_t^{z^\phi}$ of z_t according to Equation 9.

5: For each time step t, calculate the posterior $z_t^{posterior}$ of latent variable z_t according to Equation 10.

6: For each time step t, calculate the prior z_t^{prior} of latent variable z_t according to Equation 11.

7: For each time step t, decode the stock movement $\hat{y}_t$ according to Equation 12.

8: // **Temporal Attention Prediction (Section 4.3)**

9: Leverage the decoder hidden states $\boldsymbol{H}^{dec}$ to calculate the information score vector q^{dec}, the dependency score vector k^{dec}, and the normalized attention weight vector w^{dec} according to Equation 13.

10: Predict the target stock movement $\hat{y}_T$ according to Equation 14.

11: Calculate the final loss according to Equation 16 and Equation 19.

Appendix B. LLM Prompts

We include the full prompts here, comprising:

- A system prompt shared by $LLM_{\text{selection}}$, LLM_{temporal}, and LLM_{spatial}.
- A user template with the placeholders.
- A task prompt that asks the LLM to return a JSON object.

The prompts used for $LLM_{\text{selection}}$, LLM_{temporal}, and LLM_{spatial} are concatenated into a single user-role message shown in Table 8.

Table 8: Prompts used for $LLM_{\text{selection}}$, LLM_{temporal}, and LLM_{spatial}.

Component	Content
System prompt	As a stock trading news analyst, you are a helpful and precise assistant. Your task is to analyze the given text, including the correlation between the news and the given stock, the duration of the news impact, and the stocks that may be affected by the news. Output it in a standard parsable JSON format. Do not output extra information other than JSON.
User template	[Stock Name]{stock_name} [company]{comp_name} [News Content]{news_content} [Publish Time]{publish_time} [System Prompt]{system_prompt}

Task prompt	I need you to analyze the provided stock-related news from three aspects:
	1. Correlation between the news and the given stock: Rate the correlation on a scale of 0 to 10, where a higher score indicates a stronger correlation between the news and the given stock.
	2. Duration of the news impact: Rate the duration on a scale of 0 to 30, where a higher score indicates a longer potential duration of the news impact.
	3. The potential stocks that may be affected by the news, separated by commas, must include {stock_name} itself.
	(Please refrain from providing an analysis and simply provide the answer according to the following format.)
	Fill the results into JSON:
	{
	"Correlation": # Correlation score between the news and the stock
	"Duration": # Duration score of the news impact
	"Affected_stocks": # Potential stocks that may be affected by the news
	}

References

Noujoud Ahbali, Xinyuan Liu, Albert Nanda, Jamie Stark, Ashit Talukder, and Rupinder Paul Khandpur. Identifying corporate credit risk sentiments from financial news. In

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, pages 362–370, 2022.

Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 10–21, 2016.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Weijun Chen and Yanze Wang. Dhmoe: Diffusion generated hierarchical multi-granular expertise for stock prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11490–11499, 2025.

Braiden Coleman, Kenneth Merkley, and Joseph Pacelli. Human versus machine: A comparison of robo-analyst and traditional research analyst investment recommendations. *The Accounting Review*, 97(5):221–244, 2022.

Xuan-Hong Dang, Syed Yousaf Shah, and Petros Zerfos. "the squawk bot": Joint learning of time series and text data modalities for automated financial information filtering. *arXiv preprint arXiv:1912.10858*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yitong Duan, Weiran Wang, and Jian Li. Factorgcl: A hypergraph-based factor model with temporal residual contrastive learning for stock returns prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 173–181, 2025.

Robert D Edwards, John Magee, and WH Charles Bassetti. *Technical analysis of stock trends*. CRC Press, 2018.

Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. Enhancing stock movement prediction with adversarial training. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5843–5849, 2019.

Jeffrey A Frankel. *Financial markets and monetary policy*. MIT Press, 1995.

Jianliang Gao, Xiaoting Ying, Cong Xu, Jianxin Wang, Shichao Zhang, and Zhao Li. Graph-based stock recommendation by time-aware relational attention network. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(1):1–21, 2021.

Qiang Gao, Xinzhu Zhou, Kunpeng Zhang, Li Huang, Siyuan Liu, and Fan Zhou. Incorporating interactive facts for stock selection via neural recursive odes. *arXiv preprint arXiv:2210.15925*, 2022.

Jiaying Gong and Hoda Eldardiry. Multi-stage hybrid attentive networks for knowledge-driven stock movement prediction. In *International conference on neural information processing*, pages 501–513, 2021.

Yi-Ling Hsu, Yu-Che Tsai, and Cheng-Te Li. Fingat: Financial graph attention networks for recommending top-k profitable stocks. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):469–481, 2021.

Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 261–269, 2018.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kelvin JL Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. Learning to generate explainable stock predictions using self-reflective large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 4304–4315, 2024.

Shuqi Li, Weiheng Liao, Yuhan Chen, and Rui Yan. Pen: Prediction-explanation network to forecast stock price movement with better explainability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5187–5194, 2023.

Shuqi Li, Heyue Lin, Xin Liu, and Rui Yan. Expred: Explainable stock movement prediction via hybrid reflection and direct preference hierarchical optimization. *Artificial Intelligence*, page 104539, 2026.

Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23, 2014.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Andrew W Lo. The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management, Forthcoming*, 2004.

Di Luo, Weiheng Liao, Shuqi Li, Xin Cheng, and Rui Yan. Causality-guided multi-memory interaction network for multivariate stock price movement prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12164–12176, 2023.

Di Luo, Shuqi Li, Weiheng Liao, and Rui Yan. Magicnet: Memory-aware graph interactive causal network for multivariate stock price movement prediction. *IEEE Transactions on Knowledge and Data Engineering*, 37(4):1989–2000, 2025.

Xiang Ma, Tianlong Zhao, Qiang Guo, Xuemei Li, and Caiming Zhang. Fuzzy hypergraph network for recommending top-k profitable stocks. *Information Sciences*, 613:239–255, 2022.

Charles Nelson Miller, Richard Roll, William Taylor, et al. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417, 1970.

Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

Tao Ren, Ruihan Zhou, Jinyang Jiang, Jiafeng Liang, Qinghao Wang, and Yijie Peng. Riskminer: Discovering formulaic alphas via risk seeking monte carlo tree search. *arXiv preprint arXiv:2402.07080*, 2024.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286, 2014.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*, 2016.

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8415–8426, 2020.

Robert P Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):1–19, 2009.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 627–637, 2017.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Timm O. Sprenger and Isabell M. Welpe. News or noise? the stock market reaction to different types of company-specific news events. *SSRN Electronic Journal*, 2001.

Allan Timmermann and Clive WJ Granger. Efficient market hypothesis and forecasting. *International Journal of forecasting*, 20(1):15–27, 2004.

Hanshuang Tong, Jun Li, Ning Wu, Ming Gong, Dongmei Zhang, and Qi Zhang. Ploutos: Towards interpretable stock movement prediction with financial large language model. *arXiv preprint arXiv:2403.00782*, 2024.

Guifeng Wang, Longbing Cao, Hongke Zhao, Qi Liu, and Enhong Chen. Coupling macro-sector-micro financial indicators for learning stock representations with less uncertainty. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4418–4426, 2021a.

Meiyun Wang, Kiyoshi Izumi, and Hiroki Sakaji. Llmfactor: Extracting profitable factors through prompts for explainable stock movement prediction. *arXiv preprint arXiv:2406.10811*, 2024.

Zhicheng Wang, Biwei Huang, Shikui Tu, Kun Zhang, and Lei Xu. Deeptrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 643–650, 2021b.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. Tradingagents: Multi-agents llm financial trading framework. *arXiv preprint arXiv:2412.20138*, 2024.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *arXiv preprint arXiv:2306.05443*, 2023.

Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979, 2018.

Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang. Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2037–2045, 2021.

Shuo Yu, Hongyan Xue, Xiang Ao, Feiyang Pan, Jia He, Dandan Tu, and Qing He. Generating synergistic formulaic alpha collections via reinforcement learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5476–5486, 2023.