Multi-View Modeling for Stock Investment Risk Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Forecasting stock investment risk is crucial for effective financial decision-making. Existing research on stock risk forecasting is still limited due to the lack of large-scale datasets and standardized investment risk forecasting tasks. To address this problem, we construct a stock investment risk dataset that standardizes the stock risk forecasting task as regression and classification problem, providing a benchmark for stock investment risk forecasting. Recent works only based on time series data capture a limited aspect of historical stock price data. To address this issue, we propose a multi-view framework that leverages large language models (LLMs) and pre-trained vision models to extract complementary long-periodic patterns and short-periodic patterns from historical stock data. Additionally, we propose a strategy that maps these features back to the temporal domain, effectively preserving temporal dependencies. Experimental results on our dataset demonstrate that our proposed model outperform the competitive baselines in regression task and classification task of stock investment risk forecasting. The codes and dataset are release in `https://anonymous.4open.science/r/MultiV-RF-F87F`.

1 Introduction

Portfolio allocation is necessary for investors to maximize their profits, where investors select a subset of stocks from the investment universe and dynamically allocate capital among the selected assets to achieve periodic returns. To help the portfolio allocation, recent works focus on forecasting stock prices returns or trends using historical market data (P.H. & Rishad, 2020; Zou et al., 2022; Wang et al., 2024a), demonstrating the feasibility of prediction using machine learning methods. However, these studies neglect the downside risk and volatility of the stocks that is considered as stock investment risk. Stock risk forecasting focuses on quantifying the potential losses and uncertainties associated with holding a stock, providing essential guidance for risk management, portfolio optimization, and stable investment decisions.

Recent works have built several datasets for stock price forecasting (Farimani et al., 2021; Sinha et al., 2022). However, few works focus on stock investment risk forecasting dataset. To fill this gap, Luo & Liu (2025) quantify stock investment risks and construct a dataset of stock investment risk forecasting. However, the dataset lacks a standardized task formulation and is limited in scale. To this end, we scale up the benchmark and formulate regression and classification tasks of stock investment risk forecasting as well, which can contribute to the community of financial analysis.

Traditional stock forecasting methods based on time-series modeling primarily focus on the direct patch approach of stock time-series curves struggling to capture long temporal dependencies and periodic information (Szydłowski & Chudziak, 2024; Bui et al., 2025; Zhao et al., 2025). To overcome this limitation, we transform the stock curves into 2D matrix that can compress the long-periodic information as an additional view to the existing methods. Together with patch methods we propose a multi-view framework of extracting and modeling multi-view features of the stocks that can capture the short-periodic and long-periodic patterns of stock curves, which is shown in Figure 1.

Recently, large-scale foundation models demonstrate strong representation learning capabilities, including large language models (LLMs) such as GPT-2 and LLaMA (Radford et al., 2019; Touvron et al., 2023), pre-trained vision or language models such as MAE, BERT, and T5 (He et al., 2022; Devlin et al., 2019;

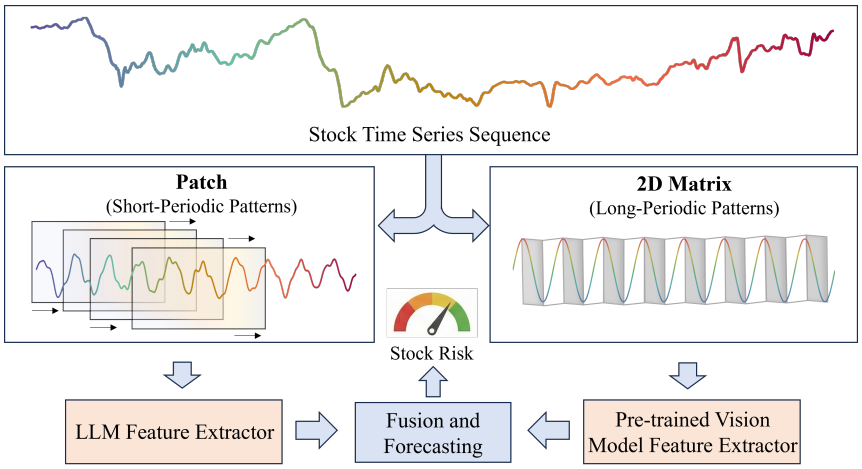


Figure 1: Multi-view modeling based on LLM feature extractor with short-periodic patterns and pre-trained vision feature extractor with long-periodic patterns.

Raffel et al., 2023), as well as multimodal foundation models such as CLIP, LLaVA, and ALIGN (Radford et al., 2021; Liu et al., 2023; Li et al., 2021). We backbone our multi-view modeling to a LLM feature extractor and pre-trained vision model to obtain short-periodic representation and long-periodic representation, respectively, which are mapped back to the temporal domain to form a unified representation for forecasting investment risk. The multi-view method significantly enhances the performance of stock investment risk forecasting. The contributions of our paper are summarized as follows:

- We build a large-scale comprehensive stock risk forecasting benchmark together with standard risk regression tasks and classification tasks, contributing to the scientific community of financial analysis.
- We propose a multi-view framework that leverages LLMs and pre-trained vision models to extract complementary short-periodic and long-periodic temporal features, providing rich representations for stock investment risk forecasting.
- We propose a fusion strategy that maps LLM representations back to the temporal domain. Our method outperforms the competitive baselines across both regression and classification tasks.

2 Related Work

2.1 Time Series Forecasting

LLM-Based The development of LLMs inspires their application in time series forecasting. Their ability to capture dependencies and encode semantic structures provides a new paradigm for time series forecasting. Methods like Zhou et al. (2023) and Jin et al. (2023) transform time series into textual representation to forecast based on LLMs' reasoning ability. Pan et al. (2025) decomposes time series into time-domain and frequency-domain components to improve forecasting performance of LLMs. Pan et al. (2024) uses semantic informed prompt learning to improve LLM's ability of time series forecasting. Chang et al. (2023) proposes a two-stage alignment framework that adapts LLMs into data-efficient time-series forecasters by combining time-series alignment with forecasting fine-tuning. Liu et al. (2024b) propose token-wise prompting to preserving the LLM's contextual learning capabilities for time series forecasting. Talukder et al. (2024) proposes a tokenized representation of time series to learn general-purpose embeddings to address time series forecasting task. Ansari et al. (2024) introduces a framework that tokenizes time series into a vocabulary and pretrains transformer-based models on these tokens to create generalizable forecasting models.

Image-Based Images have natural characteristic with time series that can provide structured representation for time series forecasting. Motivated by this observation, several recent works attempt to leverage

image-based representations to improve forecasting accuracy. Wu et al. (2022) transforms time series into multi-periodic 2D data like images to capture time series dependencies. Wang et al. (2024b) transforms time series into multi-scale frequency-based time-image to enhance the model's ability to forecast temporal patterns. Chen et al. (2024) proposes to use pre-trained visual model to enhance time series forecasting performance.

2.2 Stock Time Series Forecasting

In traditional time series forecasting, the forecast targets are included in the historical observation space. But in the financial domain, stock time series forecast extends beyond the traditional tasks involving both targets outside the historical observation space, such as stock trends and stock rankings, as well as targets within it, such as stock prices. Several recent works have been proposed to handle stock time series forecasting with different target. Li et al. (2024) proposes a market-aware transformer framework to improve the accuracy and robustness of stock price forecasting. Liu et al. (2024a) proposes a time-aware multi-view learning framework about table data to improve the accuracy of stock rank forecasting. Zhu et al. (2024) presents a trend prediction model that captures both long- and short-term temporal dependencies using an enhanced GRU architecture.

2.3 Datasets for Stock Forecasting

Several datasets have been built for stock forecasting. Xu & Cohen (2018) combines historical stock prices and related tweets to facilitate prediction of stock price movements. FNSPID (Dong et al., 2024) is a large-scale dataset providing both the news content and temporal information for stock time series forecasting and analysis. QuanSIRA(Luo & Liu, 2025) is an investment risk dataset designed to forecast the risk level of stocks. However, comprehensive datasets and standardized risk forecasting task formulation for stock risk forecasting remain scarce, motivating our construction of a new dataset about stock investment risk forecasting.

3 Stock Investment Risk Benchmark

We introduce the benchmark with risk quantification and corresponding tasks of stock investment risk forecasting.

3.1 Data Sources

We collect stock data including adjusted close price, open price, high price, low price, and trading volume, for 200 companies in the S&P 500 from 1/1/2013 to 31/12/2023 using the yfinance API.[1]

3.2 Risk Indicator

We use three normalized risk indicators: volatility, maximum drawdown, and beta coefficient to quantify the investment risk of each stock according to Luo & Liu (2025).[2]

Volatility Price volatility is used to measure the changes in stock returns. High volatility represents stock uncertainty and low volatility represents stable stock. We quantify volatility, V_t over the period $[t+1, t+\Delta d]$:

$$V_t = \sqrt{\frac{1}{\Delta d - 1} \sum_{i=t+1}^{t+\Delta d} (R_i - \bar{R})^2}, \tag{1}$$

where R_i is the return rate of a stock on i-th day:

$$R_i = \frac{p_i - p_{i-1}}{p_{i-1}}, \tag{2}$$

[1] `https://ranaroussi.github.io/yfinance/reference/index.html`.

[2] The detailed distribution of the three normalized risk indicators is in Appendix A.1.

where x_i is the closing price of a stock on i-th day, and $\bar{R}$ is the average return rate of a stock in the time $[t+1, t+\Delta d]$:

$$\bar{R} = \frac{1}{\Delta d} \sum_{i=t+1}^{t+\Delta d} R_i, \tag{3}$$

Maximum drawdown Maximum drawdown is used to measure the greatest adverse fluctuation in the stock price over a specific period. Higher maximum drawdown represents a higher level of downside risk and lower maximum drawdown represents a lower level of downside risk. We quantify maximum drawdown, M_t over the period $[t+1, t+\Delta d]$:

$$M_t = \frac{\max(P_r) - \min(P_r)}{\max(P_r)}, \tag{4}$$

where P_r is the sequential closing prices of a stock, $P_r = [p_{t+1}, p_{t+2}, \ldots, p_{t+\Delta d}]$.

Beta coefficient Beta coefficient is used to measure the discrepancy between stock returns and market. Higher beta coefficient indicates that the asset reacts more strongly to market fluctuations and lower beta coefficient suggests weaker sensitivity to market movements, meaning lower systematic risk. We quantify beta coefficient, B_t over the period $[t+1, t+\Delta d]$:

$$B_t = \frac{\mathrm{Cov}(\bar{R}, \bar{R}_m)}{\sigma_m^2}, \tag{5}$$

$$\sigma_m^2 = \frac{1}{\Delta d} \sum_{i=1}^{n} (R_i - \bar{R}_m)^2, \tag{6}$$

where $\bar{R}_m$ is the average yield rate of the market based on S&P 500.

3.3 Risk Quantification

We quantify the risk S_t in $[t+1, t+\Delta d]$ as follow:

$$S_t = \frac{1}{3}(V_t + M_t + B_t), \tag{7}$$

where a larger S_t indicates a higher risk for the corresponding stock.[3] As shown in Figure 2, the distribution of the risk values S_t shows a right-skewed and heavy-tailed behavior which is consistent with the risk distribution observed in real-world stock markets (Guo, 2017).

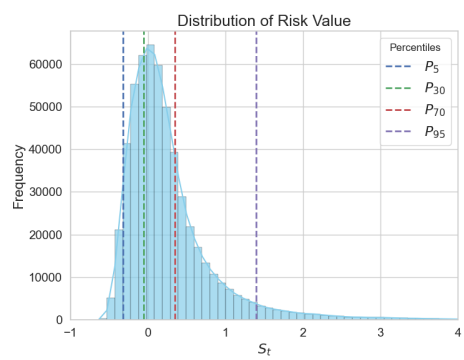


Figure 2: Distribution of the risk values S_t across all samples in the dataset.

3.4 Tasks

Stock investment risk forecasting problem are categorized into regression tasks and classification tasks. The goal is to predict the future risk of a stock over the time window $[t+1, t+\Delta d]$, based on its historical observations from the past time period $[t-w, t]$. The input is a time series feature $X \in \mathbb{R}^{w \times 1}$ from the stock historical data including adjusted close price, open price, high price, low price, or trading volume, where w is the historical window. Regression task aims to learn a regression function $f_r(\cdot)$:

$$S_t = f_{\mathrm{r}}(X). \tag{8}$$

Table 1: Statistics of the stock investment risk benchmark.

L	S	Train	Valid	Test
1	$[-1.00, -0.32]$	26,096	939	344
2	$[-0.32, -0.05]$	112,575	10,445	13,881
3	$[-0.05, 0.35]$	143,309	24,013	51,717
4	$[0.35, 1.40]$	81,743	16,597	38,561
5	$[1.40, 4.00]$	19,477	2,606	5,297
total	-	383,200	54,600	109,800

According to the distribution of S_t in Figure 2, we derive the categorical risk level L_t:

$$L_t = \begin{cases} 1, & S_t \leq P_5 \\ 2, & P_5 < S_t \leq P_{30} \\ 3, & P_{30} < S_t \leq P_{70}, \\ 4, & P_{70} < S_t \leq P_{95} \\ 5, & P_{95} < S_t \end{cases} \tag{9}$$

where L_1, L_2, L_3, L_4 and L_5 are very low, low, medium, high, and very high risk levels, respectively.[4] Classification task aims to learn a classification function $f_c(\cdot)$:

$$L_t = f_{\mathrm{c}}(X). \tag{10}$$

The two tasks are empirically hard due to implicit periodic information and correlation, where the regression is somehow difficult to be resolved compared to classification. Table 1 shows statistics of stock investment risk benchmark.[5]

4 Models

In this section, we introduce our multi-view model architecture. As shown in Figure 3, the model architecture is composed of **LLM-based Time-Series Encoder (L-TSE)**, **Visual Time-Series Encoder (V-TSE)** and **Fusion Module**. L-TSE and V-TSE are used to extracts short-periodic and long-periodic information, respectively, which are fed to fusion model to obtain a comprehensive representation.

4.1 L-TSE

As shown in the bottom of Figure 3, the L-TSE branch consists of patch construction, patch reprogramming, and feature extraction. We adopt GPT2 model (Radford et al., 2019) as L-TSE backbone to extract short-periodic pattern from stock fundamental data.

Patch Construction We segment the input sequence of stock features $X \in \mathbb{R}^{w \times 1}$ into $X_p \in \mathbb{R}^{P \times l_p}$ overlapped patches (Wu et al., 2022). The total number of patches is calculated as:

$$P = \left\lfloor \frac{w - l_p}{s} \right\rfloor + 2 \tag{11}$$

where l_p, s and w denote the patch length, stride length and the historical window, respectively. Each patch is then embedded into a d_m dimensional vector $X_d \in \mathbb{R}^{P \times d_m}$ using a linear projection layer.

[3]The detailed economic significance of S_t is in Appendix A.2.
[4]The categorization refers to Dowd & Blake (2006).
[5]In this work, we focus on single-channel time-series problem.

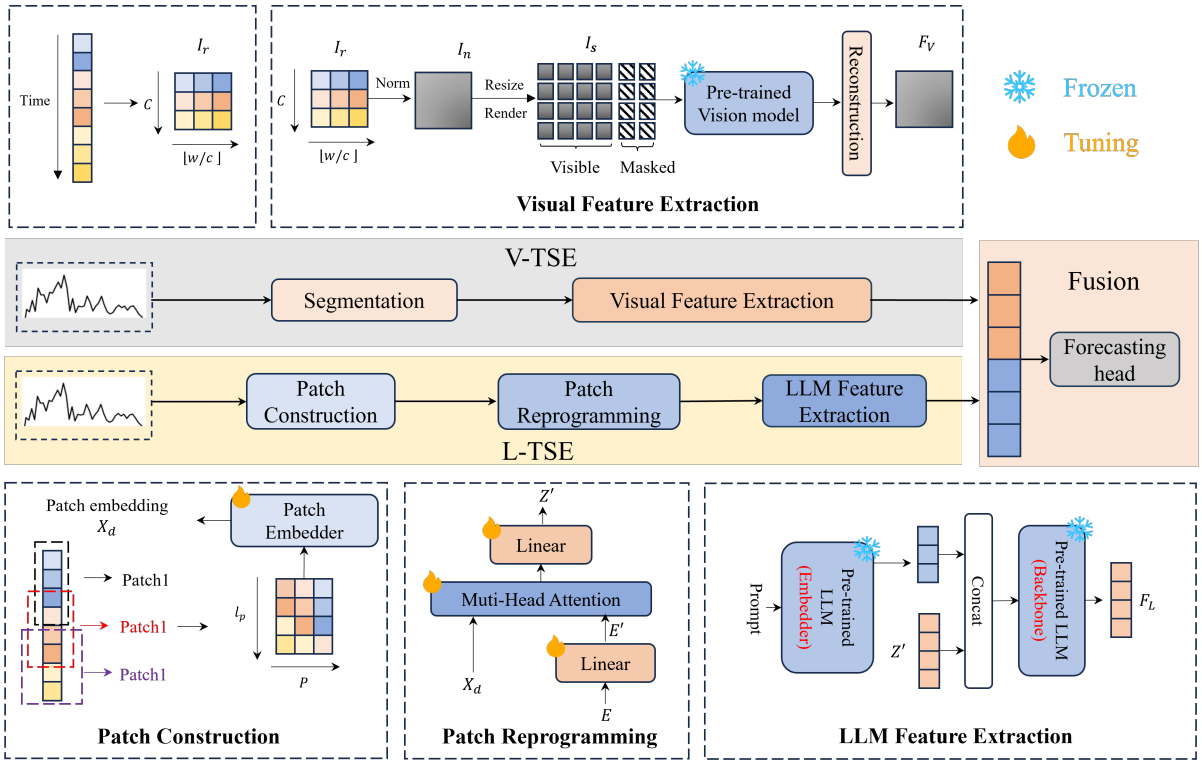


Figure 3: The multi-view modeling.

Patch Reprogramming To align patch embeddings with the representation space of the LLM, we reprogram patch embeddings through a prototype-based multi-head cross attention mechanism and project to align the LLM hidden dimensions. Directly interacting patch embeddings with the full LLM vocabulary embedding $E \in \mathbb{R}^{V \times D_l}$ computationally inefficient due to the large vocabulary size V. To alleviate this issue, we project E into a compact set of text prototypes $E' \in \mathbb{R}^{V' \times D_l}$, where $V' \ll V$ (Jin et al., 2023). The patch reprogramming is define as:

$$Z = \mathrm{MHA}(Q, K, V), \tag{12}$$

$$Q = X_d W^Q, K = E' W^K, V = E' W^V, \tag{13}$$

$$Z' = \textsc{Linear}(Z), \tag{14}$$

where $W^Q \in R^{d_m \times d_m}$ is the linear projection for the queries, and $W^K, W^V \in R^{D_l \times d_m}$ are the projections for the keys and values. $Z' \in R^{P \times D_l}$ is the aligned patch embedding. $\mathrm{MHA}(\cdot)$ is the multi-head cross attention operator and $\textsc{Linear}(\cdot)$ is a linear projection layer.

LLM Feature Extraction We design prompts based on the meta data of task description,[6] which is concatenated with patch reprogramming embedding Z' as the input to the LLM to obtain LLM feature F_L:

$$e = \textsc{Embedder}(o), \tag{15}$$

$$F_L = \textsc{Backbone}([e : Z'])[P_l :], \tag{16}$$

where o is the prompt. $e \in \mathbb{R}^{P_l \times D_l}$ is the prompt embedding by $\textsc{Embedder}(\cdot)$. $F_L \in \mathbb{R}^{P \times D_l}$ is LLM feature by $\textsc{Backbone}(\cdot)$ for stock investment risk forecasting.

[6]The detailed prompt design is in Appendix A.3.

4.2 V-TSE

As shown in the top of Figure 3, the V-TSE consists of segmentation and visual feature extraction. We employ a vision backbone initialized with a pre-trained MAE model (He et al., 2022), which enables structured representations from stock data.

Segmentation We propose to segment X into $\lfloor w/c \rfloor$ subsequences of length c (Wu et al., 2022). We segment X into 2D matrix $I_r \in R^{c \times \lfloor w/c \rfloor}$, which enables the joint modeling of intra-period variations and inter-period dependencies across the same temporal phase.

Visual Feature Extraction To align the input of pre-trained MAE model, we normalize 2D I_r to get I_n:

$$I_n = r \cdot \frac{I_r - \mu(I_r)}{\sigma(I_r)}, \tag{17}$$

where r is a hyperparameter. $\mu(\cdot)$ is mean operator and $\sigma(\cdot)$ is standard deviation operator.

We resize I_n to obtain visiable parts $I_v \in \mathbb{R}^{H \times W_v}$ to match the height and the number of visible patch columns, which is concatenated with a masked placeholder $I_m \in \mathbb{R}^{H \times W_m}$ as follows:

$$I_v = \textsc{Resize}(I_n), \tag{18}$$

$$I_f = [I_v; I_m], \tag{19}$$

where I_f is rendered to be $I_s \in \mathbb{R}^{H \times W \times 3}$ that is fed to pre-trained MAE. Finally, we reconstruct the output of the MAE to obtain the visual feature $F_V \in \mathbb{R}^{H \times W \times 3}$ as follows:

$$F_V = \textsc{Reconstruct}(\text{MAE}(I_s)), \tag{20}$$

where $\textsc{Resize}(\cdot)$ is an image rescaling operation that adjusts the spatial resolution of the input image and $\textsc{Reconstruct}(\cdot)$ is an unpatchifying operation that maps the patch space back to the image space.

4.3 Fusion Module

Given the representations of F_L from the L-TSE and F_V from the V-TSE, we project them into time-series domain to obtain $F'_L \in \mathbb{R}^{\Delta d \times 1}$ and $F'_V \in \mathbb{R}^{\Delta d \times 1}$, which are concatenated to obtain unified representation F:

$$F = [F'_L; F'_V] \in \mathbb{R}^{2\Delta d \times 1}. \tag{21}$$

The fused feature F is fed into a regression head and a classification head for forecasting stock value $\hat{S}_t$ and the risk level $\hat{L}_t$, respectively.

5 Experiments

We carry out the experiments on our proposed stock investment risk forecasting benchmark of classification task and regression task in single stock feature settings.

5.1 Settings

Baselines We compare our method with several representative time series models, including LLM-based methods TIME-LLM (Jin et al., 2023), S2IP-LLM (Pan et al., 2024), LLM-TPF (Pan et al., 2025) and the image-based methods VisionTS (Chen et al., 2024). A baseline models are trained on our dataset using their reported settings.

Data Experiments are conducted on the investment risk forecasting dataset consisting of 200 stocks covering the period from 1/1/2013 to 31/12/2023 using 6 different stock features, including adjusted price, closing price, low price, open price, trading volume, and high price.

Table 2: Results across our models and various baselines on regression task and classification task. Aclose, Close, Low, Open, Vol and High are adjusted close price, close price, low price, open price, trade volume and high price as stock features, respectively. The best scores are **bold** and the second best are underline.

		Regression Task										Classification Task									
Methods		Ours		Time-LLM		S2IP-LLM		LLM-TPF		VisionTS		Ours		Time-LLM		S2IP-LLM		LLM-TPF		VisionTS	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Aclose	96	**0.450**	**0.445**	0.521	0.474	0.563	0.487	0.655	0.507	0.573	0.498	**0.522**	**0.505**	0.516	0.458	0.481	0.452	0.471	0.319	0.446	0.360
	192	**0.468**	**0.457**	0.558	0.501	0.554	0.489	0.638	0.499	0.588	0.508	**0.506**	**0.491**	0.501	0.402	0.491	0.479	0.472	0.359	0.461	0.396
	336	0.574	0.498	0.571	0.493	**0.552**	**0.492**	0.620	0.494	0.732	0.552	0.472	0.470	0.486	0.393	**0.499**	**0.473**	0.474	0.334	0.457	0.393
Close	96	**0.453**	**0.444**	0.571	0.493	0.563	0.487	0.650	0.503	0.580	0.501	**0.525**	**0.502**	0.510	0.426	0.481	0.452	0.465	0.305	0.442	0.348
	192	**0.521**	**0.478**	0.558	0.497	0.556	0.489	0.624	0.494	0.599	0.512	**0.500**	**0.477**	0.467	0.319	0.492	0.476	0.477	0.391	0.459	0.390
	336	**0.534**	**0.488**	0.548	0.519	0.552	0.492	0.620	0.494	0.720	0.547	0.472	0.470	0.486	0.393	**0.499**	**0.473**	0.474	0.334	0.457	0.393
Low	96	**0.484**	**0.460**	0.564	0.490	0.563	0.487	0.664	0.510	0.594	0.521	**0.519**	**0.494**	0.484	0.402	0.486	0.460	0.470	0.303	0.448	0.353
	192	**0.480**	**0.466**	0.558	0.497	0.553	0.490	0.624	0.497	0.637	0.518	**0.493**	**0.487**	0.486	0.419	0.467	0.462	0.468	0.326	0.459	0.396
	336	**0.525**	**0.482**	0.548	0.521	0.547	0.497	0.619	0.494	0.718	0.550	**0.477**	**0.470**	0.468	0.318	0.471	0.301	0.472	0.342	0.438	0.384
Open	96	**0.445**	**0.443**	0.562	0.494	0.563	0.487	0.642	0.501	0.588	0.502	**0.526**	**0.502**	0.483	0.354	0.503	0.441	0.477	0.355	0.448	0.350
	192	**0.537**	**0.478**	0.585	0.494	0.552	0.491	0.627	0.496	0.617	0.511	**0.496**	**0.489**	0.407	0.401	0.470	0.464	0.473	0.370	0.456	0.393
	336	0.565	0.496	0.559	0.512	**0.551**	**0.493**	0.623	0.496	0.695	0.542	**0.492**	**0.481**	0.472	0.323	0.381	0.387	0.471	0.422	0.445	0.380
Vol	96	0.611	0.509	**0.538**	0.489	0.562	**0.487**	0.649	0.503	0.588	0.502	0.462	**0.336**	0.473	0.330	0.473	0.311	**0.473**	0.315	0.463	0.328
	192	0.599	0.504	0.559	0.491	**0.556**	**0.491**	0.646	0.502	0.624	0.516	0.472	**0.361**	0.471	0.306	**0.475**	0.342	0.473	0.319	0.457	0.351
	336	0.617	0.509	0.552	0.493	**0.550**	0.493	0.611	**0.491**	0.628	0.515	0.395	0.355	0.463	0.311	**0.477**	**0.412**	0.470	0.325	0.458	0.365
High	96	**0.439**	**0.442**	0.564	0.490	0.564	0.487	0.649	0.503	0.584	0.506	0.510	**0.493**	**0.512**	0.420	0.500	0.452	0.471	0.305	0.447	0.347
	192	**0.482**	**0.462**	0.599	0.494	0.551	0.491	0.640	0.500	0.604	0.510	**0.505**	**0.490**	0.484	0.411	0.479	0.467	0.470	0.311	0.456	0.393
	336	**0.492**	**0.479**	0.548	0.505	0.549	0.494	0.602	0.489	0.714	0.545	0.490	0.472	0.471	0.308	**0.501**	**0.482**	0.479	0.344	0.451	0.368

Implementation Details All experiments are conducted on a single NVIDIA RTX 4090 GPU with 24GB of memory. The historical window is selected from $\{96, 192, 336\}$ and the forecasting horizon for stock investment risk is $\Delta d = 30$. In the patch construction, we set path length $l_p = 16$ and stride length $s = 8$ to construct patch and we set the hyperparameter $d_m = 512$. In the segmentation, we set subsequence length $c = 7$ inspiring by the inherent periodic patterns in stock market. In the visual feature extraction, the hyperparameter r is fixed at 0.4. We use GPT-2-base as the backbone for the language model and MAE-base as the backbone for the vision model. The output dimensions of GPT-2 and MAE are denoted as $D_l = D_v = 768$.

Metrics We evaluate regression performance using Mean Squared Error (MSE) and Mean Absolute Error (MAE), while classification performance is measured by accuracy and F1 score.

5.2 Results

As shown in Table 2, our model achieves superior performances on both regression task and classification task, demonstrating a clear overall advantage. Compared with LLM-based forecasting approaches and vision-based method, our method demonstrates clear performance gains, highlighting the limitation of relying single view on long-periodic patterns or short-periodic patterns for stock risk forecasting.

Regression Task Results In the regression task, our model achieves the best performance on adjusted price, close price, low price, and high price across all historical windows. In particular, the best performance (MSE = 0.439, MAE = 0.442) is obtained when using the high price as the stock feature with a historical window of 96. Compared with LLM-based and vision-based baselines, including Time-LLM, S2IP-LLM, LLM-TPF, and VisionTS, our approach achieves significant improvements, outperforming their best results in all experiment settings by 18.7%, 19.7%, 27.1% and 23.4% based on MSE scores.

Classification Task Results In the classification task, our model achieves the best performance. Our model with open price as stock features and historical window of 96 achieves the best performance of 0.526 accuracy and 0.526 F1. Compared other methods, our approach achieves significant improvements, outperforming their best results in all experiment settings by 10.3%, 4.6%, 19.7% and 27.5% based on F1 scores.

Table 3: Results of the ablation study on L-TSE and V-TSE. The best results are **bold**.

Components		Historical Windows					
L-TSE	V-TSE	96		192		336	
Regression Task							
		MSE	MAE	MSE	MAE	MSE	MAE
✓	✗	0.522	0.474	0.558	0.501	**0.571**	**0.493**
✗	✓	0.574	0.498	0.588	0.508	0.732	0.552
✓	✓	**0.450**	**0.445**	**0.468**	**0.457**	0.574	0.497
Classification Task							
		Acc	F1	Acc	F1	Acc	F1
✓	✗	0.516	0.458	0.501	0.402	**0.486**	0.393
✗	✓	0.446	0.360	0.461	0.396	0.457	0.393
✓	✓	**0.522**	**0.505**	**0.506**	**0.491**	0.472	**0.470**

Table 4: Results of multi-view modeling (adjusted close as stock features with historical window 96) with different feature fusion strategies. $F = [F_L; F_V]$ and $F = [F'_L; F'_V]$ are the concatenation fusion in the LLM space and the temporal space, respectively. $F = V \rightarrow L(F_L; F_V)$ and $F = L \rightarrow V(F_L; F_V)$ are the cross-attention fusion with F_L and F_V as the primary modality, respectively. The best results are **bold**.

Feature fusion	Regression		Classification	
	MSE	MAE	Acc	F1
$F = [F_L; F_V]$	0.477	0.457	0.511	0.495
$F = V \rightarrow L(F_L; F_V)$	0.511	0.461	0.500	0.500
$F = L \rightarrow V(F_L; F_V)$	0.468	0.451	0.506	0.497
$F = [F'_L; F'_V]$	**0.450**	**0.445**	**0.522**	**0.505**

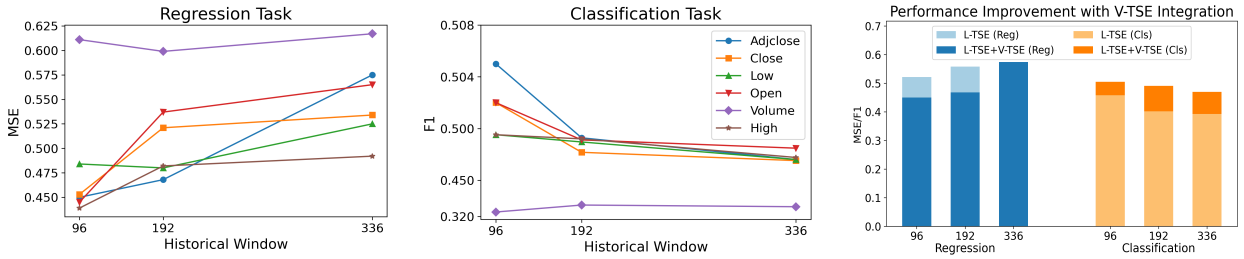
6 Analysis and Discussion

In this section, we conduct a comprehensive analysis on L-TSE and V-TSE, and the effectiveness of our fusion methods. Additionally we investigate the impact of stock features and historical windows.

6.1 L-TSE and V-TSE

To validate the effectiveness of L-TSE and V-TSE module. We perform ablation study on the two modules. As shown in Table 3, under historical window of 96 and 192, using either module alone leads to a performance drop on both regression and classification tasks. However, under the historical window of 336, incorporating both modules does not yield further performance gains compared to using L-TSE alone. In both regression and classification task, L-TSE alone ahcieves higher performance than V-TSE alone.

These results provide insights into the roles of the two modules from multi-view perspective. Although both modules are derived from the same stock price sequence, L-TSE and V-TSE focus on different views of the data. L-TSE is used to capture the short-periodic pattern and V-TSE is used to capture the long-periodic pattern. L-TSE consistently outperforms V-TSE when used alone, indicating that short-periodic information play a more dominant role in stock risk forecasting. When the historical windows becomes long, both short-periodic and long-periodic patterns are affected by increased noise, leading to performance degradation.



(a) Regression Task (b) Classification Task (c) V-TSE Improvement

Figure 4: (a)-(b):The performance of the proposed model on regression and classification tasks across different historical windows and stock features. (c) is the performance improvement of V-TSE with historical window 96 and adjusted close price as stock features. Blue bars denote regression results, while orange bars denote classification results. Light and dark colors indicate L-TSE and L-TSE+V-TSE, respectively.

6.2 Fusion Module

We conduct a systematic analysis of feature fusion (Sec. 4.3) by comparing to different fusion strategies, where we apply both cross attention and concatenation to implement the fusion module. As shown in Table 4, times-series space concatenation outperforms the high-dimensional LLM feature space fusion. Because time series fusion retains fine-grained, time-sensitive numerical patterns that are essential for stock risk forecasting. In contrast, in high dimensional LLM space, focus more on global information, making it harder to capture local price variations during fusion.

6.3 Stock Features

As shown in Figure 4a and 4b, model based on price-related features achieve better performance than trade volume based. Price-based feature provide direct information about stock return and risk-related price movements, which aligns the investment risk. In contrast, trading volume shows limited relevance to stock investment risk, since trading volume primarily reflect market attention and trading activity rather than the underlying investment risk of a stock. Trading volume contributes less discriminative information for investment risk forecasting compared to price-based features. This highlights that, for modeling investment risk, features directly related to price behavior are more reliable, while trading volume may serve better as a supplementary signal for stock investment risk forecasting.

6.4 Historical Windows

As shown in the Figure 4a and 4b, as the historical window increases, the performance of models significantly decreases (high MSE and low F1). The best performance in the regression task is obtained with the high price feature at historical windows of 96, while in the classification task, the adjusted close price feature performs best under the same historical windows. It demonstrates that massive historical information contains more long-periodic patterns that are complicated and hardly modeled, which can be recognized as noises for the stock risk modeling.

As shown in Figure 4c, with the historical windows increasing, V-TSE can obtain more long-periodic information for stock investment risk forecasting. This improvement is largely attributed to the expanded receptive field, which allows the model to integrate richer historical contexts. However, it is worth noting a saturation point in the regression task at the historical window of 336, where the L-TSE + V-TSE module yields comparable performance to the L-TSE, suggesting that the benefit of extending sequence length may diminish for specific tasks after a certain threshold.

7 Conclusion

In this paper, we construct a large-scale comprehensive stock investment risk dataset together with regression and classification tasks for benchmarking stock investment risk forecasting. To this end, we present a multi-view modeling for stock investment risk forecasting that integrates temporal information from stock prices with short-periodic feature extracted by LLM and long-periodic features from pre-trained vision models. Extensive experiments demonstrate that our method outperforms the competitive baselines, highlighting the benefit of multi-view modeling for the stock investment risk forecasting.

References

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Nguyen Kim Hai Bui, Nguyen Duy Chien, Péter Kovács, and Gergő Bognár. Transformer encoder and multi-features time2vec for financial prediction. *arXiv preprint arXiv:2504.13801*, 2025.

Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *CoRR*, 2023.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters. *arXiv preprint arXiv:2408.17253*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL `https://arxiv.org/abs/1810.04805`.

Zihan Dong, Xinyu Fan, and Zhiyuan Peng. Fnspid: A comprehensive financial news dataset in time series. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4918–4927, 2024.

Kevin Dowd and David Blake. After var: the theory, estimation, and insurance applications of quantile-based risk measures. *Journal of Risk and Insurance*, 73(2):193–229, 2006.

Saeede Anbaee Farimani, Majid Vafaei Jahan, Amin Milani Fard, and Gholamreza Haffari. Leveraging latent economic concepts and sentiments in the news for market prediction. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10. IEEE, 2021.

Zi-Yi Guo. Heavy-tailed distributions and risk management of equity market tail events. *Journal of Risk & Control*, 4(1):31–41, 2017.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. Master: Market-guided stock transformer for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 162–170, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Ying Liu, Cai Xu, Long Chen, Meng Yan, Wei Zhao, and Ziyu Guan. Table: Time-aware balanced multi-view learning for stock ranking. *Knowledge-Based Systems*, 303:112424, 2024a.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37: 122154–122184, 2024b.

Zheyang Luo and Jiangming Liu. Quansira: The quantitative investment risk modeling in stock markets with pre-trained language models. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2025. doi: 10.1109/IJCNN64981.2025.11227374.

Qihong Pan, Haofei Tan, Guojiang Shen, Xiangjie Kong, Mengmeng Wang, and Chenyang Xu. Llm-tpf: Multiscale temporal periodicity-semantic fusion llms for time series forecasting. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 6030–6038, 2025.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. S2 ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Haritha P.H. and Abdul Rishad. An empirical examination of investor sentiment and stock market volatility: evidence from india. *Financial Innovation*, 6(1):34, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL `https://arxiv.org/abs/1910.10683`.

Ankur Sinha, Satishwar Kedas, Rishu Kumar, and Pekka Malo. Sentfin 1.0: Entity-aware sentiment analysis for financial news. *Journal of the Association for Information Science and Technology*, 73(9):1314–1335, 2022.

Kamil Szydłowski and Jarosław A Chudziak. Hidformer: Transformer-style neural network in stock price forecasting. *arXiv preprint arXiv:2412.19932*, 2024.

Sabera Talukder, Yisong Yue, and Georgia Gkioxari. Totem: Tokenized time series embeddings for general time series analysis. *arXiv preprint arXiv:2402.16412*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL `https://arxiv.org/abs/2302.13971`.

Jujie Wang, Jing Liu, and Weiyi Jiang. An enhanced interval-valued decomposition integration model for stock price prediction based on comprehensive feature extraction and optimized deep learning. *Expert Systems with Applications*, 243:122891, 2024a.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. *arXiv preprint arXiv:2410.16032*, 2024b.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Yumo Xu and Shay B. Cohen. Stock movement prediction from tweets and historical prices. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1970–1979, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1183. URL `https://aclanthology.org/P18-1183`.

Cheng Zhao, Junyi Cai, and Shuyi Yang. A hybrid stock prediction method based on periodic/non-periodic features analyses. *EPJ Data Science*, 14(1):1, 2025.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Peng Zhu, Yuante Li, Yifan Hu, Qinyuan Liu, Dawei Cheng, and Yuqi Liang. Lsr-igru: Stock trend prediction based on long short-term relationships and improved gru. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, pp. 5135–5142, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704369. doi: 10.1145/3627673.3680012. URL `https://doi.org/10.1145/3627673.3680012`.

Jinan Zou, Qingying Zhao, Yang Jiao, Haiyao Cao, Yanxi Liu, Qingsen Yan, Ehsan Abbasnejad, Lingqiao Liu, and Javen Qinfeng Shi. Stock market prediction via deep learning techniques: A survey. *arXiv preprint arXiv:2212.12717*, 2022.

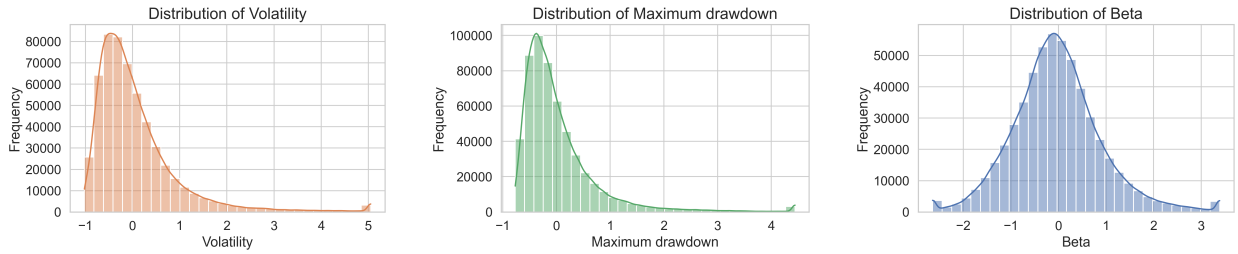
A Appendix

A.1 The distribution of risk indicators

The distributions of volatility, maximum drawdown and beta are shown in Figure 5. Due to the heavy-tailed nature of financial risk indicators, extreme values are clipped using percentiles to preserve the overall distribution.

A.2 Economic significance

The quantified investment risk S_t integrates three dimensions of investment risk: return uncertainty, downside loss severity, and systematic market exposure. Specifically, volatility captures fluctuations in returns and reflects the uncertainty, maximum drawdown measures the magnitude of extreme losses and characterizes downside risk under adverse market conditions, and the beta coefficient quantifies a stock's sensitivity to overall market movements and represents systematic risk. S_t offers a practical and actionable metric for risk monitoring, enabling the identification of potential market stress events and informing quantitative strategies.



(a) Volatility (b) Maximum drawdown (c) Beta

Figure 5: Distributions of the three risk indicators after clipping the extreme 0.5% of values.

A.3 Prompt of LLM

As shown in Figure 6, the prompt include dataset description, task description and input information of dataset and the stock name. Task description is used to describe our task. Input information includes the information about min value, max value, median value and trend.

<Dataset description>:
STOCK is a collection of daily data from S&P 500, which describes the risk of different stock risk. The stock is APPL.

<Task Description>
Forecast the next 30 steps stock risk given the previous 96 steps information.

<Input Information>
Min value:{min value}, Max value:{max value}, Median value :{median value}, Trend{Up or Down}……

Figure 6: The prompt include dataset description, task description and input information. The highlight in red is calculated by different input.