Same Company, Same Signal: The Role of Identity in Earnings Call Transcripts

Anonymous ACL submission

Abstract

Post-earnings volatility prediction is critical for investors, with previous works often leveraging earnings call transcripts under the assumption that their rich semantics contribute significantly. To further investigate how transcripts impact volatility, we introduce DEC, a dataset featuring accurate volatility calculations enabled by the previously overlooked `beforeAfterMarket` attribute and dense ticker coverage. Unlike established benchmarks, where each ticker has only around two earnings, DEC provides 20 earnings records per ticker. Using DEC, we reveal that post-earnings volatility undergoes significant shifts, with each ticker displaying a distinct volatility distribution. To leverage historical post-earnings volatility and capture ticker-specific patterns, we propose two training-free baselines: *Post-earnings Volatility* (PEV) and *Same-ticker Post-earnings Volatility* (STPEV). These baselines surpass all transcripts-based models on DEC as well as on established benchmarks. Additionally, we demonstrate that current transcript representations predominantly capture ticker identity rather than offering financially meaningful insights specific to each earnings. This is evidenced by two key observations: earnings representations from the same ticker exhibit significantly higher similarity compared to those from different tickers, and predictions from transcript-based models show strong correlations with prior post-earnings volatility.

1 Introduction

Post-earnings volatility prediction is crucial for investors and an emerging trend in the field of financial natural language processing (FinNLP). Volatility, defined as the standard deviation of returns over a specific period—post earnings call in this context—is a key financial metric for evaluating a company's performance.

Traditional finance methods primarily rely on volatility time series and statistical techniques such as GARCH and its variants (Engle, 1982; Bollerslev, 1986). However, with the rapid advancements in natural language processing (NLP) and audio processing, numerous studies have focused on utilizing unstructured earnings call data, such as transcripts and audio recordings, to enhance post-earnings volatility prediction (Qin and Yang, 2019; Yang et al., 2020; Li et al., 2020). In this pursuit, researchers have employed a variety of techniques, including heterogeneous graphs (Sawhney et al., 2020a; Liu et al., 2024b), language model pre-training (Yang et al., 2022; Niu et al., 2023) and Large Language Models (LLMs) (Cao et al., 2024a,b), to better address this complex problem.

Delving deeper into the background of earnings calls, we found that previous benchmarks, EC (Qin and Yang, 2019) and MAEC (Li et al., 2020), have overlooked a crucial attribute: `beforeAfterMarket`, which indicates whether earnings are released before the market opens or after it closes. This attribute is indispensable for accurately calculating volatility. We also observed that EC and MAEC prioritize ticker[1] coverage breadth over density, as each ticker appears around twice in these datasets. This limitation prevents tracking a company's earnings over the long term. To address this, we curated a dense earnings call dataset, DEC, where each ticker is represented with 20 earnings records. This enables robust long-term trend analysis and detailed quarter-to-quarter comparisons.

On DEC, we observe that post-earnings returns—and consequently, volatility[2]—are significantly higher than during normal periods, and that each ticker exhibits distinct post-earnings volatility patterns, we thus hypothesize that historical post-earnings volatility plays a dominant role in volatility prediction. To this end, we introduce two training-free baselines: PEV *(Post-earnings Volatility)* and STPEV *(Same-ticker Post-earnings*

[1] In this work, "ticker" refers to a company.

[2] The volatility calculation is detailed in Section 3.

Volatility). Remarkably, even with a simple mean-based implementation, our approach achieves state-of-the-art (SOTA) performance on all datasets: EC, MAEC and DEC, compared to transcripts-based models. Through further comparisons at both the representation level and the prediction level, we find that transcripts from the same company exhibit high similarity, and the predictions of transcript-based models strongly correlate with those of STPEV(Mean). This suggests that *transcripts primarily reflect ticker identity and prior post-earnings volatility distribution*, challenging the mainstream assumption that each earnings call provides financially meaningful semantics.

Our contributions include:

- We curated a dense dataset, DEC, where each ticker includes 20 earnings, in contrast to the approximately two earnings per ticker in established datasets. Additionally, DEC incorporates the previously omitted `beforeAfterMarket` attribute, enabling accurate volatility calculations.

- We propose two training-free baselines, PEV and STPEV, which achieve SOTA performance on EC, MAEC, and DEC, surpassing transcripts-based models.

- Through representation-level comparisons between examples from the same company and those across all companies, as well as prediction-level comparisons between STPEV and transcript-based models, we find that: ***transcripts mostly reflect ticker identity***.

2 Related Work

Considerable research efforts have been dedicated to leveraging earnings call transcripts, often in combination with other modalities such as audio recordings or time-series data, to model financial risk.

Transcripts-based models. A few models rely exclusively on transcripts for volatility prediction. For instance, the Multi-Round QA Attention model (Ye et al., 2020) extracts semantic information from each question-answer round and integrates features across multiple granularities to predict volatility.

Transcripts are often combined with audio recordings during earnings calls. The Multimodal Deep Regression Model (MDRM) (Qin and Yang, 2019) integrates transcript and audio information to forecast volatility. Building on MDRM, the Hierarchical Transformer-based Model (HTML) (Yang et al., 2020) employs a hierarchical transformer framework to enhance performance. Addressing the limitations of traditional language models in processing numerical information, which is critical in transcripts, Numerical HTML (NumHTML) (Yang et al., 2022) was developed. NumHTML improves predictive accuracy by incorporating numerical data, leveraging different categories of numbers and their magnitudes to augment the textual model's efficacy.

Additionally, VolTAGE (Sawhney et al., 2020a) and ECHO-GL (Liu et al., 2024b) demonstrate that correlations between stocks are beneficial for predicting volatility. These models derive stock relationships from the rich semantic content of earnings calls using a heterogeneous graph learning.

In the era of LLM, RiskLabs (Cao et al., 2024a) utilizes LLMs to encode transcripts and news articles, combining these with other modalities to deliver a comprehensive approach for volatility prediction. The ECC Analyzer (Cao et al., 2024b) employs LLMs to first extract paragraph-level general information by summarizing the text and subsequently identifies fine-grained focus sentences using Retrieval-Augmented Generation (RAG).

Liu et al. (2024a) highlight that existing pre-trained embedding models and LLM embeddings often fail to capture subtle shifts in financial narratives for the same company across different periods. To address this limitation, a specifically tailored LLM-augmented pipeline has been developed for financial semantic textual similarity.

In this work, we use different LLM embeddings to represent both vanilla transcripts and LLM fine-grained transcripts, which provide deeper insights and a more nuanced understanding.

Time series-based models The Knowledge-enhanced Financial Volatility Prediction (KeFVP) model (Niu et al., 2023) demonstrates the advantages of integrating time-series data with textual information. *Pre-earnings volatility series*, is processed by the Autoformer (Wu et al., 2021). The resulting representations are then conditionally combined with transcript representations, enhancing the model's predictive capabilities.

In this work, we introduce a second type of volatility series: *the post-earnings volatility series*, which captures the sequence of volatility observed after a prior earnings announcement and before the subsequent one. In contrast, *the pre-earnings volatility series* refers to the sequence of volatility recorded during the period leading up to a specific earnings announcement.

3 Post Earnings Volatility Prediction

Earnings call transcripts Earnings call transcripts are written records of the earnings calls held by companies at the end of each quarter or fiscal year. These transcripts capture the detailed discussions about financial results, company performance, and future projections provided by the company's executives, as well as the question and answer session with analysts and investors.

Volatility In financial terms, volatility (Kogan et al., 2009) represents the degree of variation of a trading price series over time as measured by the standard deviation of returns. Let's assume earnings is released at day t, and mathematically, volatility can be defined over a specific interval, $[t, t+\tau]$, as follows:

$$v_{[t,t+\tau]} = \log \sqrt{\sum_{i=0}^{\tau} (r_{t+i} - \overline{r})^2}, \quad (1)$$

where r_{t+i} represents the return at time $t+i$, calculated as: $r_{t+i} = \frac{C_{t+i} - C_{t+i-1}}{C_{t+i-1}}$, and C_i denotes the closing price on day i. Additionally, $\overline{r}$ is the average return over the period from t to $t+\tau$.

Volatility is a critical measure in finance as it reflects the risk associated with the price movements of a security. In previous work, Qin and Yang (2019) utilized various time intervals, $\tau = \{3, 7, 15, 30\}$, to quantify volatility, capturing both short-term and long-term market behaviors.

Volatility Time-Series The *pre-earnings volatility time-series* represents volatility from day $i - \tau + 1$ to day $i - 1$, while the *post-earnings volatility time-series* represents volatility for the $i - \tau + 1$th to the $i - 1$th earnings. The former reflects volatility leading up to an earnings announcement, while the latter captures volatility occurring between a prior and subsequent earnings announcement.

Figure 1 illustrates common approaches that leverage transcripts and volatility time series. These are processed separately, and the resulting representations are then combined for prediction.

4 A Dense Dataset: DEC

4.1 Two Widely-Used Datasets

For the task of post-earnings volatility prediction, two datasets are commonly employed: EC (Qin and Yang, 2019) and MAEC (Li et al., 2020). The MAEC dataset is further divided into two subsets, corresponding to the years 2015 and 2016.

4.2 Missing BeforeAfterMarket

During our analysis of the EC and MAEC datasets, we identified a critical oversight regarding the timing of earnings releases—specifically, the `beforeAfterMarket` attribute, which indicates whether the release occurs before or after the market opens. This attribute is essential for accurately calculating post-earnings volatility. Previous volatility calculations [3] fail to account for scenarios *where earnings are released before the market opens*, requiring the current trading day to be treated as the starting point of the volatility calculation period. Specific example is illustrated in Section A.1. Unfortunately, previous studies have often neglected this critical factor, resulting in inaccuracies in volatility measurements.

4.3 Sparse Same-ticker Representation

Tracking a company's earnings call transcripts over the long term provides valuable insights for investors by offering a deeper understanding of the company's management, strategy, and market positioning. It also reveals key performance metrics, progress on stated goals, and emerging trends that could influence future growth. Comparing transcripts over time allows investors to detect shifts in narratives or priorities, which may signal potential challenges or opportunities.

To investigate whether the current datasets supports long-termism, we define the Overlapping Earnings per Ticker (OET) as follows:

$$\text{OET} = \frac{\text{Count (testing tickers overlapped training earnings)}}{\text{Count (testing tickers)}} \quad (2)$$

This metric measures how well the training set[4] represents the testing set in terms of **ticker overlap**. Generally, a higher OET value indicates a longer-term focus on the tickers' earnings.

Table 1 shows the statistics for the EC and MAEC datasets. Regrettably, all three datasets

[3] `https://github.com/hankniu01/KeFVP/tree/main/price_data`

[4] Here, we include both the training set and validation set.

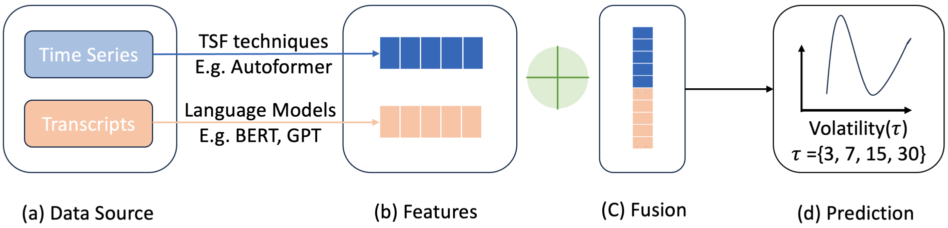


Figure 1: Overview of Post-Earnings Volatility Prediction: Time-series data is processed using time-series forecasting (TSF) techniques such as Autoformer (Wu et al., 2021), while textual data is handled by language models like BERT (Devlin, 2018) and GPT (Radford, 2018). The resulting representations are then combined for prediction.

exhibit low OET values, indicating a limited number of overlapping earnings. This limitation prevents investors from tracking a company's long-term earnings performance, which provides valuable insights into growth trends, financial health, and management effectiveness.

Dataset	Cate	# E	# T	Ratio $\frac{\#E}{\#T}$	# OE	OET
EC	All	559	272	2.055	-	-
	Train	391	243	1.609	-	-
	Val	56	56	1.0	-	-
	Test	112	112	1.0	178	1.589
MAEC-15	All	765	527	1.452	-	-
	Train	535	409	1.308	-	-
	Val	76	76	1.0	-	-
	Test	154	154	1.0	94	0.61
MAEC-16	All	1400	908	1.542	-	-
	Train	980	734	1.335	-	-
	Val	140	140	1.0	-	-
	Test	280	277	1.011	215	0.768

Table 1: Statistics for EC, MAEC-15, and MAEC-16. # E , # T and # OE defines the number of earnings, tickers, and overlapping earnings respectively. OET is the Overlapping Earnings per Ticker defined in equation 2.

4.4 A Dense Dataset: DEC

To address these limitations, we curated a new dataset: **DEC**. The DEC dataset offers four key advantages over the existing datasets:

- **Correct Volatility Calculation:** As described in Section 4.2, the `beforeAfterMarket` attribute is omitted in existing datasets. To address this, we collect this important attribute from the financial data provider EOD: [5] to ensure accurate volatility calculation.

- **Longitudinal Depth:** DEC comprises 1,800 earnings, providing a temporally dense focus on 90 specific tickers over 20 quarters, spanning the period from 2019 to 2023.

- **Latitudinal Depth:** The dataset includes representative tickers from various sectors[6] within the U.S. market, ensuring representation across a diverse range of industries.

- **Recency and Relevance:** DEC is more recent compared to existing datasets, offering up-to-date information and reflecting the latest market dynamics. Notably, it also encompasses the COVID-19 pandemic period, which exhibits distinct patterns compared to typical market conditions.

Table 2 presents the OET statistics of the DEC dataset. It is evident that the OET values increase over time, as indicated by the progression from the top-left to the bottom-right of the table. Further details regarding the curation process of the DEC dataset can be found in Appendix A.2.

DEC enables long-term trend analysis and quarter-to-quarter analysis by providing richer context for comparing sequential performance, and understanding transitions between quarters.

5 On the Importance of Past Post-earnings Volatility

5.1 Distribution Shift After Earnings

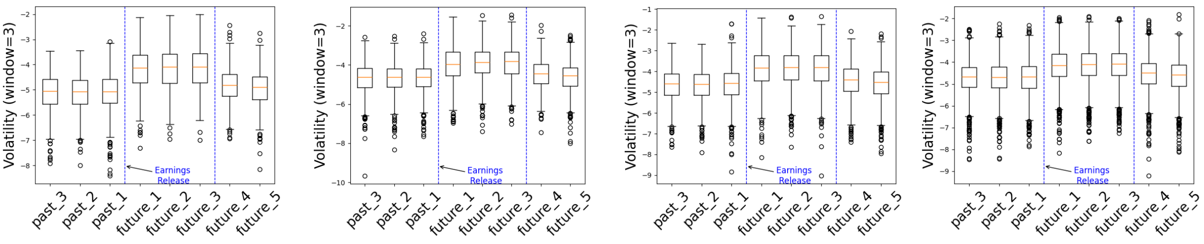
We observe a significant increase in returns following earnings announcements across the EC, MAEC, and DEC datasets. Specifically, the return on the first day after earnings r_{future_1} is consistently higher than on pre-earnings days or other subsequent post-earnings days. We hypothesize that this phenomenon stems from the market's reaction to freshly disclosed and potentially unexpected financial information from the company. Details are provided in Appendix B.1.

[5] `https://eodhd.com/`

[6] `https://seekingalpha.com/etfs-and-funds/etf-tables/key_markets`

	First Quarter			Second Quarter			Third Quarter			Fourth Quarter		
Year	Count(Training)	Count(Testing)	OET	Count(Training)	Count(Testing)	OET	Count(Training)	Count(Testing)	OET	Count(Training)	Count(Testing)	OET
2019	0	90	0	90	90	1.0	180	90	2.0	270	90	3.0
2020	360	90	4.0	450	90	5.0	540	90	6.0	630	90	7.0
2021	720	90	8.0	810	90	9.0	900	90	10.0	990	90	11.0
2022	1080	90	12.0	1170	90	13.0	1260	90	14.0	1350	90	15.0
2023	1440	90	16.0	1530	90	17.0	1620	90	18.0	1710	90	19.0

Table 2: DEC Dataset Statistics. The dataset focuses on 90 tickers in the U.S. market, spanning 20 quarters. It is *dense in ticker coverage* and OET values, defined in equation 2, increase over time.



(a) EC (b) MAEC-15 (c) MAEC-16 (d) DEC

Figure 2: Comparison of three-day volatility before and after earnings announcements. Earnings are released between the day labeled `past_1` and the day labeled `future_1`. Days where the volatility calculation involves the return of `future_1` exhibit significantly higher volatility compared to others. This pattern holds consistently across all windows {3, 7, 15, 30}. Further details are provided in Appendix B.1.

According to the definition of volatility in equation 1, we conclude that volatility calculations involving the first daily return after earnings r_{future_1} should be higher compared to periods without earnings. Given τ as the volatility window, a total of τ days of volatility are directly influenced by r_{future_1}. As illustrated in Figure 2, which compares three-day volatility before and after earnings announcements across the EC, MAEC and DEC datasets[7], post-earnings volatility within a three-day window is notably higher than the volatility observed on other trading days. This pattern remains consistent across other time windows (7, 15, and 30 days), with further details in Appendix B.1.

Given the pronounced differences in volatility distribution between post-earnings periods[8] and non-earnings periods, we hypothesis that:

Incorporating <u>past</u> post-earnings volatility is essential for volatility prediction.

5.2 Ticker-Specific Volatility Signature

It also has been observed that the post-earnings volatility for each company tends to follow a distinct distribution. As depicted in Figure 3, companies such as JNJ, V, and TSLA[9] exhibit markedly different three-day post-earnings volatility patterns.

We term this phenomenon as *Volatility Signature*, reflecting the persistence of a company's volatility patterns over time. This signature likely arises from intrinsic company characteristics that remain relatively stable over short periods. These characteristics may include industry and sector classification, operational dynamics, company size and market position, and financial structure. Motivated by the *Volatility Signature*, we refine our hypothesis:

Incorporating <u>same-ticker</u> past post-earnings volatility is critical for volatility prediction.

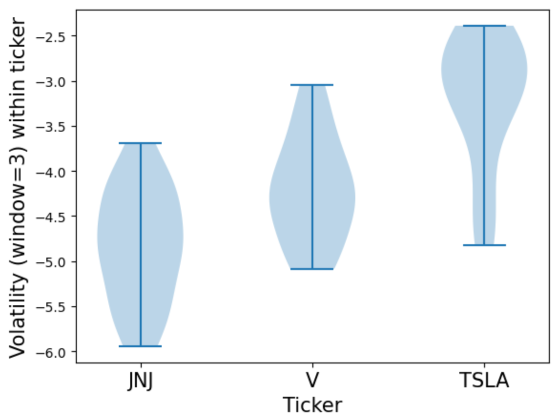


Figure 3: 3-days post earnings volatility comparison across companies: JNJ(Johnson & Johnson), V(Visa), and TSLA(Tesla). Totally 20 earnings, from 2019 to 2023, are involved for plot.

[7]The `beforeAfterMarket` attribute is adjusted based on the original EC and MAEC datasets. The plot without `beforeAfterMarket` adjustment is provided in Appendix B.1

[8]Precisely, post-earnings periods refer to the days where calculations involving the first daily return after earnings.

[9]JNJ, V, and TSLA refer to Johnson & Johnson, Visa, and Tesla, respectively.

Model	EC					MAEC-15					MAEC-16					Average
	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	
Vpast	1.12	2.99	0.83	0.42	0.23	-	-	-	-	-	-	-	-	-	-	-
Price LSTM	0.75	1.97	0.46	0.32	0.24	-	-	-	-	-	-	-	-	-	-	-
BiLSTM + ATT	0.74	1.98	0.44	0.30	0.23	0.696	1.599$^\sharp$	0.560$^\sharp$	0.339$^\sharp$	0.284$^\sharp$	0.691	1.544$^\sharp$	0.571$^\sharp$	0.362$^\sharp$	0.288$^\sharp$	0.709
HAN(Glove)	0.60	1.43	0.46	0.31	0.20	-	-	-	-	-	-	-	-	-	-	-
MDRM(Audio)	0.60	1.41	0.44	0.32	0.22	-	-	-	-	-	-	-	-	-	-	-
MDRM(Text+Audio)	0.58	1.37	0.42	0.30	0.22	0.630	1.425$^\sharp$	0.488$^\sharp$	0.320$^\sharp$	0.285$^\sharp$	0.618	1.426$^\sharp$	0.476$^\sharp$	0.311$^\sharp$	0.259$^\sharp$	0.609
HTML(Text)	0.46	1.18	0.37	0.15	0.13	0.514	1.199$^\sharp$	0.440$^\sharp$	0.231$^\sharp$	0.187$^\sharp$	0.579	1.287$^\sharp$	0.479$^\sharp$	0.300	0.249$^\sharp$	0.518
HTML(Text+Audio)	0.40	0.85	0.35	0.25	0.16	0.487	1.065$^\sharp$	0.416$^\sharp$	0.272$^\sharp$	0.196$^\sharp$	0.556	1.160$^\sharp$	0.515$^\sharp$	0.314$^\sharp$	0.236$^\sharp$	0.481
VolTAGE	0.31	0.63	0.29	0.17	0.14	-	-	-	-	-	-	-	-	-	-	-
KeFVP$^{\flat\flat}$	$\underline{0.300}$	0.610	0.291	0.183	0.114	**0.204**	0.418	0.187	0.122	0.087	0.318	0.445	0.279	0.303	0.177	0.274
SVM(TF-IDF)^b	0.70	1.70	0.50	0.34	0.25	-	-	-	-	-	-	-	-	-	-	-
bc-LSTM^b	0.59	1.42	0.44	0.30	0.22	-	-	-	-	-	-	-	-	-	-	-
Multi-Fusion CNN^b	0.41	0.73	0.35	0.29	0.28	-	-	-	-	-	-	-	-	-	-	-
NumHTML(Text+Audio)$^\natural$	0.31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ensemble(Text+Audio)^b	0.302	0.601	0.308	0.181	0.119	-	-	-	-	-	-	-	-	-	-	-
RiskLabs†	0.324	0.585	0.317	0.233	0.171	-	-	-	-	-	-	-	-	-	-	-
ECC Analyzer ‡	0.314	0.553	0.306	0.237	0.158	-	-	-	-	-	-	-	-	-	-	-
GPT4o Pred (3 shot) $^\alpha$	0.609	1.433	0.501	0.26	0.244	0.345	0.585	0.404	0.22	0.169	0.441	0.545	0.538	0.51	0.171	0.465
Gemini Pred (3 shot) $^\alpha$	0.592	1.368	0.487	0.251	0.263	0.451	0.824	0.598	0.219	0.163	0.337	0.475	0.404	0.3	0.171	0.466
Vanilla(Voyage) $^\alpha$	0.387	0.751	0.375	0.245	0.177	0.34	0.623	0.303	0.232	0.201	0.274	0.458	0.267	0.222	0.151	0.334
Vanilla(Gecko) $^\alpha$	0.36	0.664	0.356	0.237	0.182	0.283	0.513	0.27	0.19	0.159	0.254	0.402	0.25	0.249	0.116	0.299
Vanilla(OpenAI) $^\alpha$	0.339	0.682	0.311	0.209	0.155	0.319	0.596	0.290	0.206	0.184	$\underline{0.235}$	0.402	0.232	0.177	0.127	0.298
GPT4o(Summarization) $^\alpha$	0.299	0.585	0.283	0.188	0.142	0.3	0.548	0.274	0.204	0.174	0.246	0.405	0.236	0.221	0.124	0.282
GPT4o(Task-Specific) $^\alpha$	0.314	0.624	0.294	0.195	0.142	0.268	0.505	0.248	0.164	0.155	0.232	0.372	0.232	0.215	0.111	0.271
Gemini(Summarization) $^\alpha$	0.275	0.568	0.244	0.167	0.122	0.268	0.494	0.254	0.165	0.158	0.23	0.372	0.236	0.202	0.109	$\underline{0.258}$
Gemini(Task-Specific) $^\alpha$	0.284	0.583	0.252	0.175	0.128	0.276	0.508	0.262	0.176	0.158	0.235	0.383	0.229	0.212	0.115	0.265
PEV(Mean) $^\alpha$	0.399	0.743	0.389	0.262	0.201	0.305	0.532	0.301	0.209	0.177	$\underline{0.23}$	0.38	0.229	0.173	0.139	0.311
STPEV(Mean) $^\alpha$	0.349	0.724	0.33	0.205	0.138	0.301	0.571	0.273	0.19	0.17	0.271	0.459	0.273	0.209	0.144	0.307
PEV(Mean)(Aug) $^\alpha$	0.367	0.712	0.351	0.235	0.17	0.283	0.514	0.271	0.188	0.157	**0.229**	0.37	0.24	0.168	0.139	0.293
STPEV(Mean)(Aug) $^\alpha$	**0.296**	0.569	0.293	0.201	0.122	$\underline{0.225}$	0.443	0.214	0.13	0.112	0.25	0.5	0.237	0.159	0.103	**0.257**

Table 3: The overall performance on EC, MAEC-15 and MAEC-16 datasets.The models below the double line and marked with α are implemented in this work. The results with $\natural$, $\sharp$, b, $\flat\flat$, $\dagger$ and $\ddagger$ are retrieved from Yang et al. (2022), Li et al. (2020), Sawhney et al. (2020b), Niu et al. (2023), Cao et al. (2024a) and Cao et al. (2024b) respectively, and the remainder are from Sawhney et al. (2020a). The best results are in bold, and the second-best results are underlined. To ensure a fair comparison, the `beforeAfterMarket` is not adjusted for the results presented here.

5.3 Simple Baselines

PEV Building on our analysis, we propose a training-free baseline for post-earnings volatility prediction, referred to as the *Post-Earnings Volatility* (PEV) model. The PEV(X) model primarily utilizes *historical post-earnings volatility* as input and applies an aggregation function, X, which we instantiate as the mean in this work.

STPEV To incorporate the concept of *Volatility Signature*, we introduce a refined variant of the PEV, termed the *Same-Ticker Post-Earnings Volatility* (STPEV). This variant exclusively leverages *historical post-earnings volatility from the same ticker*, thereby improving its capacity to capture the characteristics specific to each company.

6 Main Results

6.1 Evaluations on EC and MAEC Datasets

Augmentation on EC and MAEC datasets. As shown in Table 1, EC, MAEC-15 and MAEC-16 datasets all suffer from a limited number of same-ticker earnings, which restricts STPEV's ability to fully exploit the ticker-specific volatility signature. To mitigate this limitation, we augment these datasets by extending their historical range from the past 5 years.[10]. Appendix C contrasts the data statistics between the original and augmented datasets. By augmenting the datasets, we achieve higher OET values, enhancing the suitability of the datasets for both PEV and STPEV.

Baselines To validate the effectiveness of PEV and STPEV, we benchmark their performance against several established methods. These baseline methods include MDRM (Text+Audio) (Qin and Yang, 2019), HTML (Text+Audio) (Yang et al., 2020), VolTAGE (Sawhney et al., 2020a), NumHTML (Text+Audio) (Yang et al., 2022), KeFVP (Niu et al., 2023), RiskLabs (Cao et al., 2024a), and ECC Analyzer (Cao et al., 2024b).

In addition to previous works, we also implement several transcripts-based baselines:

- **LLM Direct Prediction:** We prompt the LLMs[11] using few-shot learning with the task description and the previous *(earnings call transcripts, volatility)* pairs to directly predict the volatility. Details are in Appendix E.3.
- **Vanilla Text:** Earnings call transcripts are directly processed to generate text embed-

[10] We only extend the price records, such as close price, daily return, and volatility, without the earnings call transcripts.

[11] GPT4o-2024-08-06 and Gemini-1.5-Flash

dings using various models, including OpenAI embeddings, Gecko embeddings (Lee et al., 2024), and a financial-domain-specific model, Voyage embeddings[12]. These embeddings are processed by a simple 2-layer MLP model, which is also used for LLM fine-grained text.

- **LLM Fine-Grained Text:** Earnings call transcripts are scrutinized using LLMs, specifically GPT-4o and Gemini-1.5-Flash, employing two distinct strategies: *Summarization* and *Task-Specific*, to generate fine-grained summaries, which are then used to obtain embeddings[13]. Details, such as prompt templates, can be found in Appendix E.3.

PEV and STPEV settings. We evaluate PEV and STPEV with the following two simple implementations: PEV(Mean) and STPEV(Mean).

Analysis. As shown in Table 3, direct predictions from LLMs exhibit the worst performance, with MSEs of 0.465 and 0.466 for GPT4o and Gemini, respectively, highlighting the limited ability of LLMs to effectively handle regression tasks.

The vanilla transcripts using OpenAI and Gecko embeddings outperform all previous works except KeFVP (Niu et al., 2023), demonstrating the superior capability of LLM-based embeddings. Utilizing LLMs to process and analyze transcripts further improves performance, achieving the best MSE of 0.258 with the Gemini(summarization) strategy.

The PEV(Mean) and STPEV(Mean) achieve MSEs of 0.311 and 0.307 on original datasets, which are reasonable given the scarcity of prior earnings, with details in Appendix C. However, when applying the PEV(Mean) and STPEV(Mean) to the augmented datasets, the MSEs decrease to 0.293 and 0.257, with the latter achieving SOTA.

6.2 Evaluations on DEC Dataset

Evaluation settings. Following Section 6.1, we evaluate various transcript-based models on DEC. These include LLM few-shot direct prediction, using 8 prior *(earnings call transcripts, volatility)* pairs, and LLM embeddings, comprising two vanilla transcript embeddings[14] and four fine-grained text embeddings. Additionally, PEV(Mean) and STPEV(Mean) are applied to the DEC dataset.

To explore the role of semantics in transcripts and how ticker identity is purely reflected in transcript-level representations[15], we introduce two random embeddings[16]:

- **Random(All):** Each transcript is assigned a random embedding. This approach effectively removes both semantics and ticker identity.
- **Random(Ticker):** Transcripts belonging to the same ticker are assigned the same randomly generated embedding. This removes semantics while preserving ticker identity.

We only report the results from 2021 to 2023 in Table 4, as we suffer from sparse overlapping earnings for the years 2019 and 2020. The full results can be found in Appendix E.

Performance of different models. As shown in Table 4, direct prediction by GPT-4o and Gemini yield the worst performance with the MSEs of 0.345 and 0.339. Transcripts-based models exhibit comparable performance among themselves: the best model, Gemini(Summarization), and the worst model, Vanilla(Gecko), achieve average MSEs of 0.260 and 0.269, respectively. Further analysis in Appendix E.4 shows that two strategies fail to instruct LLMs to generate distinct texts for the same transcripts, whereas different LLMs can differentiate the same transcripts effectively.

Despite containing no semantics or insights in their input, the Random(Ticker) embeddings and STPEV(Mean), with overall MSEs of 0.265 and 0.252 respectively, surpass various types of semantically meaningful transcripts models. Moreover, models that lack ticker identity, such as Random(All) embeddings and the PEV(Mean), with average MSEs of 0.323 and 0.307 respectively, consistently underperform compared to other models that include ticker identity across all quarters.

These findings suggest that *the ticker identity plays a determinant role*, while *the semantic content of transcripts is plausibly not useful*. This motivates us to raise the following hypothesis:

Are the semantics of transcripts converging toward ticker identity?

[12] Specifically, OpenAI text-embedding-3-large model, Text-embedding-005 model, and Voyage-Finance-2 model.

[13] OpenAI embeddings are used for fine-grained texts.

[14] Voyage has been shown to perform poorly in Table 3.

[15] The main difference between PEV(Mean) and STPEV(Mean) lies in the ticker identity, which we aim to embed solely at the "transcript-level" representation.

[16] Random embeddings are also fed into the 2-layer MLP.

Year	Model	First Quarter					Second Quarter					Third Quarter					Fourth Quarter					Average
		$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	
2021	GPT4o Pred (8 shot)	0.306	0.713	0.235	0.152	0.124	0.357	0.707	0.237	0.187	0.297	0.439	0.742	0.414	0.318	0.284	0.293	0.653	0.247	0.153	0.12	0.349
	Gemini Pred (8 shot)	0.273	0.583	0.243	0.146	0.121	0.327	0.563	0.223	0.204	0.317	0.471	0.972	0.44	0.289	0.181	0.347	0.696	0.329	0.199	0.163	0.354
	Vanilla (OpenAI)	0.170	0.357	0.148	0.079	0.097	0.250	0.457	0.212	0.145	0.185	0.372	0.547	0.462	0.285	0.194	**0.213**	0.501	0.173	0.109	0.068	0.251
	Vanilla (Gecko)	0.200	0.419	0.191	0.104	0.087	0.269	0.464	0.245	0.176	0.189	0.350	0.523	0.377	0.291	0.211	0.253	0.535	0.223	0.161	0.094	0.268
	GPT4o (Summarization)	0.183	0.390	0.162	0.097	0.084	0.277	0.482	0.252	0.171	0.204	0.353	0.549	0.430	0.278	0.156	0.234	0.544	0.190	0.124	0.079	0.262
	GPT4o (Task-specific)	0.177	0.388	0.164	0.088	0.070	**0.246**	0.444	0.214	0.145	0.180	0.357	0.515	0.428	0.295	0.189	0.242	0.537	0.197	0.145	0.088	0.255
	Gemini (Summarization)	0.175	0.356	0.153	0.083	0.106	**0.246**	0.454	0.212	0.144	0.173	**0.322**	0.502	0.394	0.240	0.151	0.255	0.553	0.215	0.142	0.109	0.249
	Gemini (Task-specific)	0.176	0.384	0.159	0.094	0.067	0.249	0.435	0.210	0.156	0.197	0.347	0.503	0.407	0.286	0.190	0.248	0.548	0.215	0.140	0.088	0.255
	Random (All)	0.249	0.472	0.231	0.150	0.143	0.294	0.497	0.291	0.189	0.201	0.433	0.603	0.512	0.359	0.259	0.300	0.577	0.280	0.212	0.132	0.319
	Random (Ticker)	0.190	0.380	0.171	0.112	0.096	0.275	0.449	0.246	0.180	0.226	0.381	0.555	0.414	0.321	0.236	0.255	0.535	0.224	0.163	0.097	0.275
	PEV(Mean)	0.216	0.433	0.209	0.115	0.105	0.271	0.451	0.239	0.184	0.209	0.405	0.580	0.429	0.342	0.270	0.288	0.568	0.260	0.199	0.127	0.295
	STPEV(Mean)	**0.156**	0.368	0.149	0.067	0.041	0.249	0.463	0.209	0.150	0.173	0.333	0.525	0.353	0.260	0.196	0.222	0.536	0.177	0.114	0.062	**0.240**
2022	GPT4o Pred (8 shot)	0.416	0.964	0.348	0.214	0.137	0.39	0.814	0.368	0.239	0.141	0.349	0.85	0.308	0.162	0.076	0.316	0.683	0.268	0.203	0.109	0.368
	Gemini Pred (8 shot)	0.467	1.013	0.43	0.235	0.19	0.326	0.682	0.283	0.212	0.127	0.34	0.808	0.299	0.154	0.099	0.297	0.69	0.248	0.17	0.078	0.357
	Vanilla (OpenAI)	**0.258**	0.523	0.253	0.160	0.095	0.354	0.671	0.251	0.260	0.235	0.261	0.622	0.237	0.112	0.072	0.256	0.585	0.195	0.156	0.088	0.282
	Vanilla (Gecko)	0.302	0.603	0.293	0.188	0.125	0.353	0.659	0.325	0.238	0.188	0.265	0.608	0.234	0.134	0.083	0.274	0.588	0.228	0.178	0.101	0.298
	GPT4o (Summarization)	0.332	0.641	0.311	0.225	0.149	0.317	0.656	0.273	0.174	0.164	**0.239**	0.586	0.195	0.107	0.069	0.291	0.629	0.217	0.201	0.118	0.295
	GPT4o (Task-specific)	0.309	0.585	0.286	0.220	0.147	**0.307**	0.618	0.196	0.193	0.220	0.243	0.578	0.219	0.110	0.065	0.275	0.599	0.209	0.194	0.099	0.284
	Gemini (Summarization)	0.297	0.591	0.269	0.208	0.119	0.348	0.660	0.240	0.244	0.246	0.252	0.605	0.220	0.109	0.073	**0.250**	0.573	0.180	0.151	0.094	0.287
	Gemini (Task-specific)	0.291	0.599	0.276	0.169	0.120	0.314	0.656	0.254	0.187	0.157	0.243	0.584	0.216	0.109	0.065	0.267	0.604	0.191	0.174	0.098	0.279
	Random (All)	0.316	0.614	0.326	0.203	0.121	0.410	0.746	0.390	0.291	0.213	0.324	0.674	0.298	0.195	0.131	0.324	0.597	0.304	0.231	0.163	0.343
	Random (Ticker)	0.270	0.553	0.269	0.159	0.098	0.308	0.609	0.235	0.223	0.163	0.255	0.610	0.230	0.113	0.068	0.285	0.602	0.242	0.187	0.108	0.279
	PEV(Mean)	0.326	0.619	0.326	0.215	0.146	0.380	0.719	0.343	0.272	0.185	0.310	0.647	0.285	0.185	0.121	0.316	0.618	0.283	0.220	0.143	0.333
	STPEV(Mean)	0.270	0.584	0.245	0.152	0.099	0.310	0.640	0.270	0.201	0.129	0.243	0.592	0.219	0.104	0.057	0.278	0.599	0.236	0.183	0.095	**0.275**
2023	GPT4o Pred (8 shot)	0.34	0.9	0.217	0.146	0.097	0.307	0.775	0.225	0.133	0.097	0.324	0.613	0.325	0.211	0.147	0.301	0.627	0.259	0.177	0.143	0.318
	Gemini Pred (8 shot)	0.325	0.825	0.25	0.124	0.103	0.296	0.731	0.222	0.141	0.089	0.32	0.627	0.355	0.172	0.128	0.277	0.602	0.247	0.153	0.105	0.305
	Vanilla (OpenAI)	0.268	0.663	0.197	0.109	0.104	0.274	0.643	0.235	0.134	0.084	0.226	0.439	0.249	0.129	0.088	0.258	0.557	0.222	0.142	0.111	0.257
	Vanilla (Gecko)	0.257	0.659	0.189	0.104	0.076	0.266	0.619	0.226	0.132	0.089	0.215	0.408	0.213	0.126	0.113	**0.229**	0.473	0.208	0.131	0.105	**0.242**
	GPT4o (Summarization)	0.262	0.642	0.202	0.108	0.097	0.266	0.634	0.233	0.121	0.076	0.220	0.405	0.226	0.144	0.103	0.250	0.542	0.214	0.145	0.098	0.249
	GPT4o (Task-specific)	0.249	0.634	0.188	0.097	0.078	0.262	0.621	0.221	0.125	0.080	0.220	0.418	0.209	0.142	0.110	0.253	0.542	0.220	0.140	0.110	0.246
	Gemini (Summarization)	0.262	0.629	0.194	0.113	0.114	0.260	0.624	0.213	0.129	0.074	**0.210**	0.415	0.224	0.119	0.083	0.238	0.523	0.210	0.127	0.092	0.243
	Gemini (Task-specific)	0.262	0.637	0.204	0.115	0.092	0.269	0.622	0.228	0.136	0.088	**0.210**	0.408	0.205	0.126	0.102	0.244	0.528	0.211	0.133	0.105	0.246
	Random (All)	0.317	0.729	0.248	0.163	0.130	0.352	0.752	0.290	0.213	0.152	0.280	0.481	0.265	0.193	0.182	0.279	0.566	0.251	0.155	0.143	0.307
	Random (Ticker)	0.247	0.633	0.182	0.095	0.080	0.255	0.596	0.208	0.130	0.088	0.228	0.438	0.212	0.133	0.130	0.238	0.513	0.203	0.124	0.110	**0.242**
	PEV(Mean)	0.309	0.725	0.236	0.150	0.124	0.330	0.723	0.279	0.186	0.134	0.262	0.463	0.249	0.172	0.166	0.278	0.584	0.245	0.148	0.134	0.295
	STPEV(Mean)	**0.239**	0.611	0.180	0.093	0.074	**0.253**	0.601	0.209	0.122	0.081	0.227	0.432	0.215	0.132	0.130	0.246	0.520	0.215	0.133	0.118	**0.242**

Table 4: The overall performance on DEC.

Ticker Identity	Model	Cosine Similarity	
		Within-Ticker	All-dataset
With	Vanilla (OpenAI)	0.9	0.7
	Vanilla (Gecko)	0.958	0.865
	GPT4o (Summarization)	0.92	0.685
	GPT4o (Task-specific)	0.929	0.724
	Gemini (Summarization)	0.931	0.713
	Gemini (Task-specific)	0.918	0.728
	Random (Ticker)	1.0	0.752
	Average	**0.937**	**0.738**
Without	Random (All)	0.765	0.753

Table 5: The mean cosine similarity between the within-ticker group and the all-dataset group.

Representation-level comparisons between within-ticker group and all-dataset group. We compare the cosine similarity (for each earnings record) within individual tickers and across the entire dataset for text embeddings. As shown in Table 5, the within-ticker similarity is consistently higher than the overall similarity when ticker identity is present, even for texts scrutinized by LLMs. This finding aligns with Liu et al. (2024a). For more details, please refer to Appendix E.5.

This observation is intuitive, as earnings calls for each company tend to follow similar and structured patterns over time. This finding supports the hypothesis that *transcripts primarily capture the ticker identity*, owing to the inherently similar nature of transcripts for the same ticker.

Prediction-level comparisons between transcripts-based models and STPEV(Mean). We also compare the predictions of different transcripts-based models with those of the STPEV(Mean) and calculate the Pearson correlation coefficients between them. As shown in Table 6, The predictions of transcripts-based models and STPEV(Mean) are highly linearly correlated for all models that contain the ticker identity, with an average correlation coefficient of 0.847 over 3 years. This also validates our hypothesis that *transcripts approximate the prior same-ticker post-earnings volatility*, where we simply use the mean value to represent the distribution. Details are in Appendix E.6.

Ticker Identity	Model	Yearly Average		
		2021	2022	2023
With	Vanilla (OpenAI)	0.87	0.873	0.866
	Vanilla (Gecko)	0.753	0.825	0.81
	GPT4o (Summarization)	0.799	0.792	0.848
	GPT4o (Task-specific)	0.831	0.855	0.916
	Gemini (Summarization)	0.852	0.854	0.867
	Gemini (Task-specific)	0.816	0.843	0.867
	Random (Ticker)	0.786	0.925	0.946
	Average	**0.815**	**0.852**	**0.874**
Without	Random (All)	0.183	0.007	0.004

Table 6: Pearson correlation coefficients between the predictions of transcripts models and of STPEV(Mean).

7 Conclusion

In this work, we introduce a dense earnings call dataset: DEC. Motivated by the dominant role of prior post-earnings volatility on DEC, we propose two training-free baselines, PEV and STPEV, which surpass various transcripts-based models. We further confirm that transcripts primarily capture ticker identity and approximate the past post-earnings volatility distribution, by representation-level and prediction-level comparisons.

Limitations

Limited transcript representations. The current methods used to process transcripts are relatively simplistic. Incorporating additional numeric data, such as actual EPS and revenue, along with analysts' expectations for these metrics, could enhance the quality and depth of representations.

Lack of incorporation of more price information. This work utilizes only post-earnings volatility. Combining post-earnings and pre-earnings volatility could further improve prediction accuracy.

Acknowledgements

We thank EOD[17] for providing reliable earnings information and price records. We also thank Seeking Alpha[18] for supplying sector-level information and earnings call transcripts.

References

Victor L. Bernard and Jacob K. Thomas. 1989. Post-earnings-announcement drift: Delayed price response or risk premium? *Journal of Accounting Research*, 27:1–36.

Tim Bollerslev. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327.

Yupeng Cao, Zhi Chen, Qingyun Pei, Fabrizio Dimino, Lorenzo Ausiello, Prashant Kumar, KP Subbalakshmi, and Papa Momar Ndiaye. 2024a. Risklabs: Predicting financial risk using large language model based on multi-sources data. *arXiv preprint arXiv:2404.07452*.

Yupeng Cao, Zhi Chen, Qingyun Pei, Nathan Jinseok Lee, KP Subbalakshmi, and Papa Momar Ndiaye. 2024b. Ecc analyzer: Extract trading signal from earnings conference calls using large language model for stock performance prediction. *arXiv preprint arXiv:2404.18470*.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. 2023. Tsmixer: An all-mlp architecture for time series forecasting.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Robert F Engle. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007.

[17]https://eodhd.com/
[18]https://seekingalpha.com/

Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. 2009. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280, Boulder, Colorado. Association for Computational Linguistics.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. 2024. Gecko: Versatile text embeddings distilled from large language models.

Jiazheng Li, Linyi Yang, Barry Smyth, and Ruihai Dong. 2020. Maec: A multimodal aligned earnings conference call dataset for financial risk prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3063–3070, New York, NY, USA. Association for Computing Machinery.

Jiaxin Liu, Yi Yang, and Kar Yan Tam. 2024a. Beyond surface similarity: Detecting subtle semantic shifts in financial narratives. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2641–2652, Mexico City, Mexico. Association for Computational Linguistics.

Mengpu Liu, Mengying Zhu, Xiuyuan Wang, Guofang Ma, Jianwei Yin, and Xiaolin Zheng. 2024b. Echo-gl: Earnings calls-driven heterogeneous graph learning for stock movement prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(12):13972–13980.

Hao Niu, Yun Xiong, Xiaosu Wang, Wenjing Yu, Yao Zhang, and Weizu Yang. 2023. KeFVP: Knowledge-enhanced financial volatility prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11499–11513, Singapore. Association for Computational Linguistics.

Yu Qin and Yi Yang. 2019. What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 390–401, Florence, Italy. Association for Computational Linguistics.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Ramit Sawhney, Piyush Khanna, Arshiya Aggarwal, Taru Jain, Puneet Mathur, and Rajiv Ratn Shah. 2020a. VolTAGE: Volatility forecasting via text audio fusion with graph convolution networks for earnings calls. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8001–8013, Online. Association for Computational Linguistics.

Ramit Sawhney, Puneet Mathur, Ayush Mangal, Piyush Khanna, Rajiv Ratn Shah, and Roger Zimmermann. 2020b. Multimodal multi-task financial risk forecasting. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 456–465, New York, NY, USA. Association for Computing Machinery.

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

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*.

Linyi Yang, Jiazheng Li, Ruihai Dong, Yue Zhang, and Barry Smyth. 2022. Numhtml: Numeric-oriented hierarchical transformer model for multi-task financial forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11604–11612.

Linyi Yang, Tin Lok James Ng, Barry Smyth, and Ruihai Dong. 2020. Html: Hierarchical transformer-based multi-task learning for volatility prediction. In *Proceedings of The Web Conference 2020*, pages 441–451.

Zhen Ye, Yu Qin, and Wei Xu. 2020. Financial risk prediction with multi-round q&a attention network. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4576–4582. International Joint Conferences on Artificial Intelligence Organization. Special Track on AI in FinTech.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR.

A Dataset Details

A.1 Missing BeforeAfterMarket: an example

Previous studies have overlooked the fact that earnings can be released *before the market opens*, which is 9 AM in U.S. exchanges. Here is a specific example in EC dataset: Target (TGT) Q3 2017 Earnings Call[19]. This earnings release occurred at 8:00 AM ET on Nov. 15, 2017. Since market participants had access to all earnings disclosures during trading hours on Nov. 15, the first post earnings day should be considered as Nov. 15 rather than Nov. 16. According to the definition of post-earnings volatility, the three-day volatility should account for the trading days *{Nov. 15, Nov. 16, and Nov. 17}*, not *{Nov. 16, Nov. 17, and Nov. 20}*. Consequently, the volatility should be recalculated as -2.726, whereas previous studies incorrectly recorded it as -3.703.

Regrettably, both the EC and MAEC datasets exhibit critical errors when calculating volatility for earnings released before the market opening. Specifically, the EC, MAEC15, and MAEC16 datasets contain 368, 395, and 584 earnings released before market opening, accounting for 69.3%, 64.3%, and 64.2% of the datasets, respectively. Thus, we believe that the volatility values used in prior studies are unreliable.

A.2 DEC Dataset Details

To ensure a diverse representation of tickers in the U.S. markets, we selected 11 sectors: Technology, Healthcare, Industrial, Utility, Real Estate, Basic Materials, Financial Services, Consumer Discretionary, Consumer Staples, Communication Services, and Energy. Each sector exhibits distinct characteristics and patterns in response to earnings calls, driven by differences in business models, investor expectations, and macroeconomic influences.

For each sector, we selected companies that are among the top 10 holdings in sector-specific ETFs. For example, in the technology sector, the top 10 companies held by the XLK ETF[20] include Apple Inc., NVIDIA Corp., Microsoft Corp., Broadcom Inc., Salesforce Inc., Oracle Corp., Cisco Systems Inc., Adobe Inc., Accenture PLC Class A, and Advanced Micro Devices Inc. In this way, we further ensure that DEC includes most of the representative companies in the U.S. while maintaining diversity.

[19] https://seekingalpha.com/article/4125212-target-tgt-q3-2017-results-earnings-call-transcript

[20] https://seekingalpha.com/symbol/XLK

After identifying 110 tickers, we merged earnings call transcripts with price records. During this process, some tickers were excluded due to various reasons, such as incomplete earnings cycles (fewer than 20 earnings records) or missing price data or the `beforeAfterMarket` attribute from EOD.

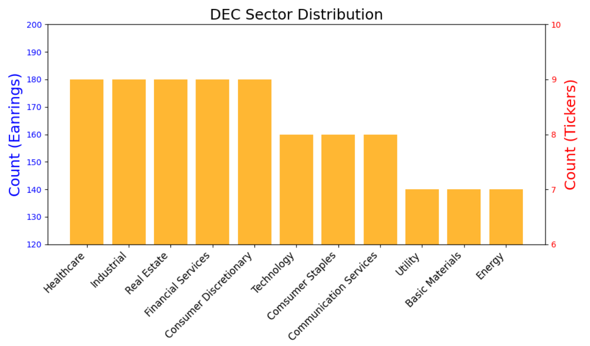


Figure 4: DEC Sector Distribution

Ultimately, we retained 90 tickers across 11 sectors, with each ticker containing 20 earnings records spanning from the first quarter of 2019 to the last quarter of 2023, resulting in a total of 1,800 earnings records. The sector distribution is illustrated in Figure 4.

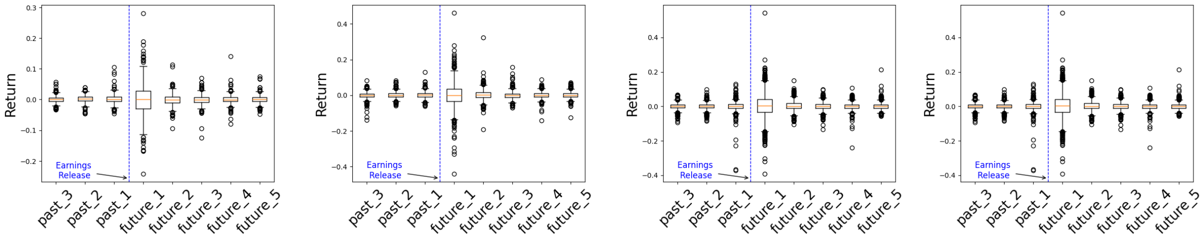
B Two Observations from DEC

B.1 Post Earnings Volatility Distribution Drift

Figure 5 compares the daily returns before and after earnings for the EC, MAEC, and DEC datasets. The return on the first day after earnings, r_{future_1}, is significantly higher than on other days. In contrast, Figure 6 compares the daily returns *without beforeAfterMarket adjustment* from the original EC and MAEC datasets[21]. In this case, the effect tends to diminish or disappear due to the incorrect time used in identifying r_{future_1}. This observation further validates the importance of the `beforeAfterMarket` attribute.

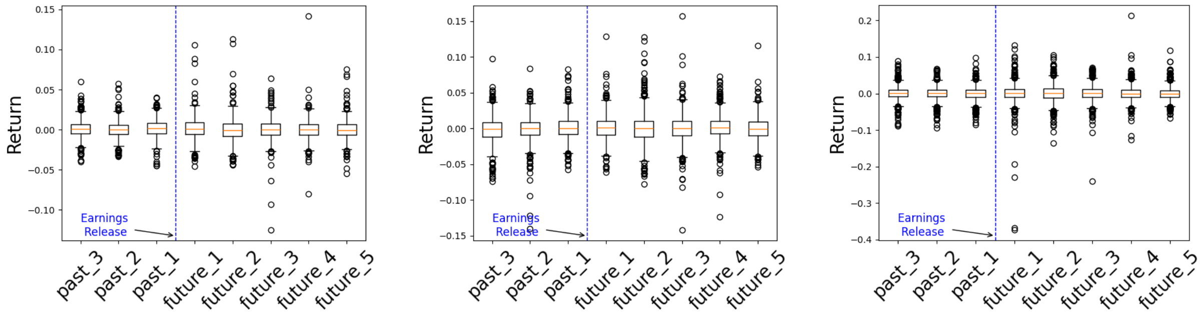
In Section 5.1, we conclude that days involving the volatility calculation of r_{future_1}(the first day after earnings) exhibit significantly higher volatility compared to other days. Figure 2 illustrates this pattern for a 3-day window. In contrast, Figure 7 compares the same trend *without the beforeAfterMarket adjustment* using the original EC and MAEC datasets, where the increased volatility within the 3-day window is observed to dilute. This

[21] https://github.com/hankniu01/KeFVP/tree/main/price_data



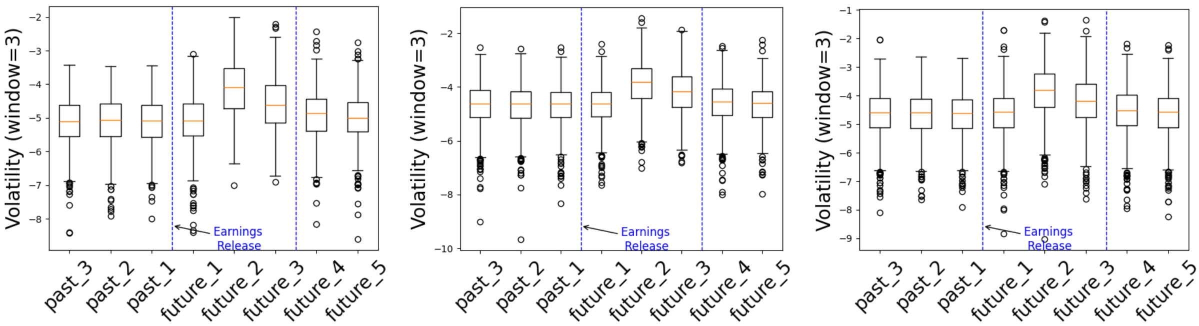
(a) EC (b) MAEC-15 (c) MAEC-16 (d) DEC

Figure 5: Comparison of **returns** before and after earnings announcements. Earnings are released between the day labeled `past_1` and the day labeled `future_1`. The return on `future_1` (the first day after earnings) is significantly higher than on other days, a phenomenon known as PEAD (Bernard and Thomas, 1989).



(a) EC (b) MAEC-15 (c) MAEC-16

Figure 6: Comparison of **returns (w/o beforeAfterMarket adjustment)** before and after earnings announcements. Earnings are released between the day labeled `past_1` and the day labeled `future_1`.



(a) EC (b) MAEC-15 (c) MAEC-16

Figure 7: Comparison of 3-day volatility **(w/o beforeAfterMarket adjustment)** before and after earnings announcements. Earnings are released between the day labeled `past_1` and the day labeled `future_1`.

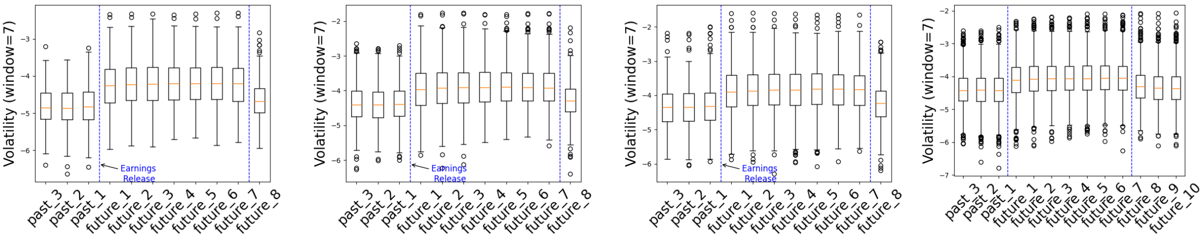
further highlights the importance of incorporating the `beforeAfterMarket` attribute.

Furthermore, Figure 8 demonstrates that the post-earnings volatility distribution drift persists for a 7-day window. In contrast, Figure 9 presents the same comparison *without the beforeAfterMarket adjustment*, where the phenomenon diminishes at the beginning and end of the volatility window. This further underscores the importance of incorporating the `beforeAfterMarket` attribute.

We do not present volatility comparisons for window sizes of 15 and 30, as they could not fit into a single figure due to space constraints. Nevertheless, the same distribution drift is observed for these window sizes.

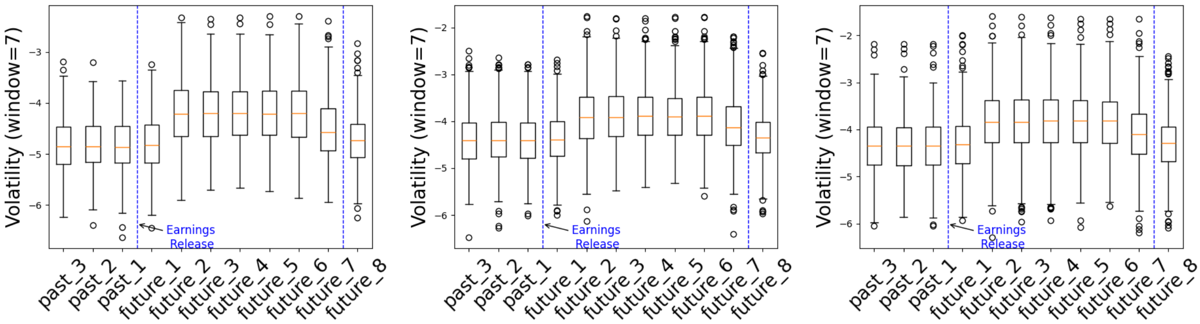
B.2 Ticker-Specific Volatility Regime

The ticker-specific volatility regime posits that each company tends to follow a distinct post-earning volatility distribution. We term this phenomenon as *Volatility Signature*, which likely arises from intrinsic company characteristics that remain relatively stable over short periods. These characteris-



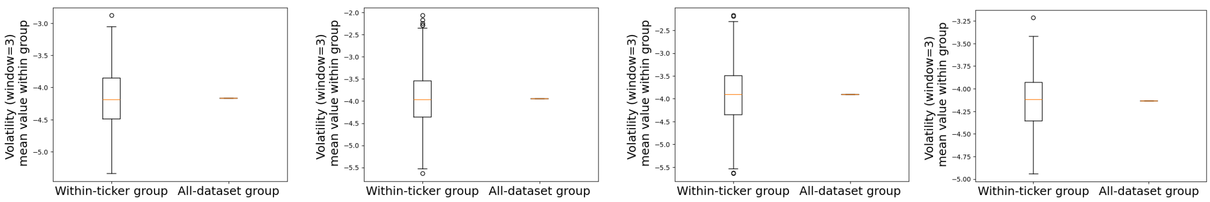
(a) EC (b) MAEC-15 (c) MAEC-16 (d) DEC

Figure 8: Comparison of seven-day volatility before and after earnings announcements. Earnings are released between the day labeled `past_1` and the day labeled `future_1`. Days where the volatility calculation involves the return of `future_1` exhibit significantly higher volatility compared to others.



(a) EC (b) MAEC-15 (c) MAEC-16

Figure 9: Comparison of seven-day volatility **(w/o beforeAfterMarket adjustment)** before and after earnings announcements. Earnings are released between the day labeled `past_1` and the day labeled `future_1`.



(a) EC (b) MAEC-15 (c) MAEC-16 (d) DEC

Figure 10: Comparison of the mean of three-day volatility between the within-ticker group and the all-dataset group.

tics may include industry and sector classification, operational dynamics, company size and market position, and financial structure. Such stable properties act as anchors, mitigating extreme volatility fluctuations and maintaining predictable patterns of post-earnings volatility, even in response to periodic financial disclosures.

To further illustrate the volatility signature, Figure 10 compares the mean values of three-day volatility between the within-ticker group and the all-dataset group. This analysis is particularly relevant, as the mean function is primarily used and benchmarked against baselines in Section 6. The figure reveals significant variation in mean values across tickers (as shown on the y-axis), underscoring the motivation for introducing STPEV as an enhancement to PEV.

C Augmentation on EC and MAEC

Dataset	EC		MAEC-15		MAEC-16	
Type	Original	Augmented	Original	Augmented	Original	Augmented
Range	2017-2017	2012-2017	2015-2015	2010-2015	2015-2016	2011-2016
Count (Train)	179	2195	94	3192	215	5765
Count (Test)	112	112	154	154	280	280
OET	1.598	19.775	0.61	20.727	0.768	20.812

Table 7: EC and MAEC statistics between the original and the augmented for STPEV. OET is defined as the proportion of overlapping training earnings over testing tickers defined in equation 2.

Since the PEV and STPEV take historical post-earnings volatility as input, the current EC and MAEC datasets, which lack sufficient previous same-ticker earnings records, must be left-extended to earlier years. Table 7 compares the data statistics of testing tickers overlapped training earnings relative to testing tickers (OET) before and after augmentation. It is evident that the OET values are significantly improved by left-extending the datasets.

D Implementation Details for Transcripts-based Models

One NVIDIA L40 GPU is used for the transcript-based models. The learning rate is set to 1e-4, with a batch size of 32 and a random seed of 2021. The models are trained for up to 10 epochs using early-stopping techniques. The embedding dimensions for OpenAI, Gecko, and Voyage embeddings are 3071, 756, and 1024, respectively. The 2-layer MLP has a hidden size of 512 in the middle layer. All results are based on a single run.

E Detailed Evaluations and Analysis on DEC

E.1 Full Results for Transcripts-based Models.

Due to the limited number of overlapping earnings in 2019 and 2020 within the DEC dataset[22], which affects the suitability of the STPEV baseline, we present results only for the years 2021 to 2023 in Section 6.2. Here, we provide the complete results on DEC, including three additional STPEV variants: STPEV(Median), STPEV(LR), and STPEV(MLP). The training-free median implementation is similar to the mean approach. In contrast, the linear regression and MLP methods require training and parameter selection. To address this, we use cross-validation for each quarter, as earnings released in the same quarter share the same number of prior post-earnings volatility.

As shown in Table 8, transcript-based models and STPEV(Mean) underperform STPEV(LR) during the early years of 2019 and 2020, where the limited number of previous same-ticker earnings makes it challenging to capture the prior post-earnings volatility distribution. On the other hand, relatively complex implementations of STPEV, such as STPEV(LR) and STPEV(MLP), underperform simpler approaches like STPEV(Mean) and STPEV(Median) when sufficient previous same-ticker earnings are available.

[22] In real-world applications, it is typically possible to gather sufficient historical earnings data.

E.2 Pre-earnings Volatility Series Prediction.

Since volatility prediction is a type of time-series forecasting problem, we follow KeFVP (Niu et al., 2023) and predict volatility using pre-earnings volatility series (with a window size of 22) through various time-series forecasting models, including DLinear (Zeng et al., 2023), TSMixer (Chen et al., 2023), TimeNet (Wu et al., 2022), and FEDformer (Zhou et al., 2022). Table 9 compares these time-series forecasting models with STPEV(Mean). We observe that simpler models, such as DLinear and TSMixer, outperform other forecasting models, though their performance and that of STPEV(Mean) vary across different years. This suggests that using pre-earnings volatility time series is also an effective approach for post-earnings volatility prediction. Future research could explore combining pre-earnings and post-earnings volatility series to achieve improved prediction accuracy.

E.3 Transcripts-based Models with LLM

LLM Direct Prediction We utilize few-shot learning to prompt LLMs for direct volatility prediction, providing the task description and prior *(earnings call transcripts, volatility)* pairs within the prompt. For EC and MAEC, three randomly selected pairs are used as demonstrations, while for DEC, eight ticker-specific prior pairs are included as demonstrations. The prompt template is illustrated in Figure 11.

LLM Fine-grained Text We also leverage LLMs to extract signals and insights from vanilla earnings call transcripts. Specifically, we deploy two distinct prompt strategies:

- *Summarization:* We prompt the LLMs to extract key points from the transcripts, focusing on different perspectives such as financial performance metrics, management commentary, and operational updates. The prompt template is illustrated in Figure 12.
- *Task-Specific:* Involves providing the LLMs with the context of the post-earnings volatility prediction task, requiring them to generate insightful comments tailored to this objective. The prompt template is illustrated in Figure 13.

E.4 Clustering among Different Texts

Given the extremely similar performances of different transcript-based models, we further compare

Year	Model	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Average
		$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	
2019	Vanilla (OpenAI)	-	-	-	-	-	0.550	0.693	0.763	0.446	0.296	0.258	0.563	0.245	0.130	0.092	0.266	0.466	0.263	0.196	0.140	0.358
	Vanilla (Gecko)	-	-	-	-	-	0.376	0.563	0.385	0.287	0.268	0.316	0.603	0.270	0.219	0.173	0.241	0.425	0.217	0.170	0.154	0.311
	GPT4o (Summarization)	-	-	-	-	-	0.317	0.596	0.398	0.168	0.104	0.279	0.585	0.272	0.155	0.104	0.247	0.433	0.227	0.171	0.159	0.281
	GPT4o (Task-specific)	-	-	-	-	-	0.328	0.606	0.416	0.180	0.109	0.285	0.605	0.278	0.154	0.104	0.231	0.407	0.214	0.162	0.142	0.281
	Gemini (Summarization)	-	-	-	-	-	0.349	0.646	0.444	0.193	0.112	0.277	0.585	0.268	0.153	0.102	0.230	0.409	0.218	0.157	0.138	0.285
	Gemini (Task-specific)	-	-	-	-	-	0.324	0.617	0.408	0.173	0.099	0.291	0.596	0.280	0.170	0.119	0.233	0.404	0.223	0.161	0.143	0.283
	Random (All)	-	-	-	-	-	0.326	0.603	0.376	0.197	0.129	0.358	0.623	0.328	0.288	0.191	0.276	0.448	0.256	0.206	0.192	0.320
	Random (Ticker)	-	-	-	-	-	0.878	1.733	0.389	0.442	0.947	0.322	0.599	0.310	0.225	0.154	0.228	0.396	0.214	0.158	0.145	0.476
	PEV(Mean)	-	-	-	-	-	0.266	0.522	0.288	0.149	0.107	0.294	0.574	0.279	0.188	0.133	0.221	0.394	0.208	0.151	0.130	0.260
	STPEV(Mean)	-	-	-	-	-	0.415	0.902	0.442	0.198	0.118	0.299	0.638	0.295	0.159	0.105	0.272	0.523	0.265	0.181	0.119	0.329
	STPEV(LR)	-	-	-	-	-	**0.227**	0.488	0.254	0.108	0.059	**0.229**	0.522	0.241	0.102	0.051	**0.200**	0.383	0.195	0.135	0.086	**0.219**
	STPEV(MLP)	-	-	-	-	-	0.648	1.122	0.665	0.459	0.348	0.305	0.671	0.330	0.150	0.068	0.259	0.504	0.259	0.174	0.099	0.404
2020	Vanilla (OpenAI)	0.783	0.744	0.447	0.646	1.295	0.339	0.637	0.321	0.235	0.162	0.328	0.684	0.245	0.209	0.177	0.325	0.540	0.275	0.284	0.200	0.444
	Vanilla (Gecko)	0.795	0.753	0.502	0.650	1.276	0.358	0.620	0.334	0.285	0.192	0.349	0.734	0.258	0.203	0.203	0.310	0.535	0.269	0.256	0.179	0.453
	GPT4o (Summarization)	0.781	0.701	0.470	0.652	1.302	0.294	0.598	0.276	0.191	0.112	0.363	0.771	0.263	0.216	0.202	0.268	0.482	0.222	0.220	0.147	0.427
	GPT4o (Task-specific)	0.787	0.733	0.479	0.648	1.286	0.342	0.619	0.329	0.261	0.159	0.311	0.690	0.209	0.169	0.176	0.280	0.508	0.225	0.234	0.153	0.430
	Gemini (Summarization)	0.787	0.679	0.463	0.666	1.340	0.306	0.610	0.294	0.207	0.114	0.369	0.769	0.271	0.239	0.196	0.277	0.483	0.223	0.233	0.170	0.435
	Gemini (Task-specific)	0.747	0.688	0.437	0.620	1.244	0.300	0.602	0.271	0.213	0.116	0.317	0.700	0.216	0.182	0.172	0.291	0.503	0.248	0.251	0.160	0.414
	Random (All)	0.909	0.842	0.563	0.751	1.481	0.393	0.625	0.359	0.345	0.244	0.372	0.760	0.299	0.237	0.191	0.351	0.546	0.288	0.340	0.232	0.506
	Random (Ticker)	0.810	0.756	0.496	0.680	1.308	0.437	0.694	0.413	0.379	0.262	0.351	0.760	0.271	0.204	0.171	0.346	0.551	0.304	0.310	0.221	0.486
	PEV(Mean)	0.750	0.772	0.504	0.631	1.095	0.381	0.623	0.350	0.314	0.239	0.349	0.738	0.276	0.212	0.171	0.348	0.576	0.304	0.298	0.216	0.457
	STPEV(Mean)	0.817	0.725	0.477	0.695	1.370	0.438	0.786	0.383	0.346	0.237	**0.269**	0.685	0.196	0.112	0.083	0.311	0.536	0.275	0.281	0.151	0.459
	STPEV(LR)	**0.422**	0.664	0.370	0.453	0.200	**0.277**	0.614	0.222	0.148	0.126	0.288	0.755	0.203	0.110	0.082	**0.259**	0.609	0.224	0.125	0.077	**0.311**
	STPEV(MLP)	0.468	0.678	0.357	0.598	0.237	0.499	0.901	0.525	0.454	0.115	0.318	0.753	0.220	0.163	0.138	0.342	0.614	0.329	0.332	0.094	0.407
2021	Vanilla (OpenAI)	0.170	0.357	0.148	0.079	0.097	0.250	0.457	0.212	0.145	0.185	0.372	0.547	0.462	0.285	0.194	**0.213**	0.501	0.173	0.109	0.068	0.251
	Vanilla (Gecko)	0.200	0.419	0.191	0.104	0.087	0.269	0.464	0.245	0.176	0.189	0.350	0.523	0.377	0.291	0.211	0.253	0.535	0.223	0.161	0.094	0.268
	GPT4o (Summarization)	0.183	0.390	0.162	0.097	0.084	0.277	0.482	0.252	0.171	0.204	0.353	0.549	0.430	0.278	0.156	0.234	0.544	0.190	0.124	0.079	0.262
	GPT4o (Task-specific)	0.177	0.388	0.164	0.088	0.070	0.246	0.444	0.214	0.145	0.180	0.357	0.515	0.428	0.295	0.189	0.242	0.537	0.197	0.145	0.088	0.255
	Gemini (Summarization)	0.175	0.356	0.153	0.083	0.106	0.246	0.454	0.212	0.144	0.173	0.322	0.502	0.394	0.240	0.151	0.255	0.553	0.215	0.142	0.109	0.249
	Gemini (Task-specific)	0.176	0.384	0.159	0.094	0.067	0.249	0.435	0.210	0.156	0.197	0.347	0.503	0.407	0.286	0.190	0.248	0.548	0.215	0.140	0.088	0.255
	Random (All)	0.249	0.472	0.231	0.150	0.143	0.294	0.497	0.291	0.189	0.201	0.433	0.603	0.512	0.359	0.259	0.300	0.577	0.280	0.212	0.132	0.319
	Random (Ticker)	0.190	0.380	0.171	0.112	0.096	0.275	0.449	0.246	0.180	0.226	0.381	0.555	0.414	0.321	0.236	0.255	0.535	0.224	0.163	0.097	0.275
	PEV(Mean)	0.216	0.433	0.209	0.115	0.105	0.271	0.451	0.239	0.184	0.209	0.405	0.580	0.429	0.342	0.270	0.288	0.568	0.260	0.199	0.127	0.295
	STPEV(Mean)	**0.156**	0.368	0.149	0.067	0.041	0.249	0.463	0.209	0.150	0.173	0.333	0.525	0.353	0.260	0.196	0.222	0.536	0.177	0.114	0.062	0.240
	STPEV(LR)	0.185	0.443	0.180	0.073	0.045	**0.191**	0.412	0.170	0.114	0.069	**0.277**	0.602	0.249	0.158	0.099	0.248	0.548	0.235	0.139	0.072	**0.225**
	STPEV(MLP)	0.230	0.537	0.194	0.125	0.063	0.402	0.458	0.182	0.142	0.827	0.315	0.537	0.362	0.161	0.200	0.268	0.527	0.220	0.167	0.159	0.304
2022	Vanilla (OpenAI)	**0.258**	0.523	0.253	0.160	0.095	0.354	0.671	0.251	0.260	0.235	0.261	0.622	0.237	0.112	0.072	0.256	0.585	0.195	0.156	0.088	0.282
	Vanilla (Gecko)	0.302	0.603	0.293	0.188	0.125	0.353	0.659	0.325	0.238	0.188	0.265	0.608	0.234	0.134	0.083	0.274	0.588	0.228	0.178	0.101	0.298
	GPT4o (Summarization)	0.332	0.641	0.311	0.225	0.149	0.317	0.656	0.273	0.174	0.164	**0.239**	0.586	0.195	0.107	0.069	0.291	0.629	0.217	0.201	0.118	0.295
	GPT4o (Task-specific)	0.309	0.585	0.286	0.220	0.147	0.307	0.618	0.196	0.193	0.220	0.243	0.578	0.219	0.110	0.065	0.275	0.599	0.209	0.194	0.099	0.284
	Gemini (Summarization)	0.297	0.591	0.269	0.208	0.119	0.348	0.660	0.240	0.244	0.246	0.252	0.605	0.220	0.109	0.073	**0.250**	0.573	0.180	0.151	0.094	0.287
	Gemini (Task-specific)	0.291	0.599	0.276	0.169	0.120	0.314	0.656	0.254	0.187	0.157	0.243	0.584	0.216	0.109	0.065	0.267	0.604	0.191	0.174	0.098	0.279
	Random (All)	0.316	0.614	0.326	0.203	0.121	0.410	0.746	0.390	0.291	0.213	0.324	0.674	0.298	0.195	0.131	0.324	0.597	0.304	0.231	0.163	0.343
	Random (Ticker)	0.270	0.553	0.269	0.159	0.098	0.308	0.609	0.235	0.223	0.163	0.255	0.610	0.230	0.113	0.068	0.285	0.602	0.242	0.187	0.108	0.279
	PEV(Mean)	0.326	0.619	0.326	0.215	0.146	0.380	0.719	0.343	0.272	0.185	0.310	0.647	0.285	0.185	0.121	0.316	0.618	0.283	0.220	0.143	0.333
	STPEV(Mean)	0.270	0.584	0.245	0.152	0.099	0.310	0.640	0.270	0.201	0.129	0.243	0.592	0.219	0.104	0.057	0.278	0.599	0.236	0.183	0.095	**0.275**
	STPEV(LR)	0.305	0.739	0.273	0.145	0.063	**0.251**	0.759	0.140	0.075	0.030	0.292	0.723	0.258	0.120	0.068	0.264	0.690	0.195	0.107	0.065	0.278
	STPEV(MLP)	0.479	1.133	0.264	0.194	0.325	0.354	0.777	0.210	0.272	0.157	0.486	0.910	0.271	0.699	0.066	0.716	1.090	0.821	0.800	0.155	0.509
2023	Vanilla (OpenAI)	0.268	0.663	0.197	0.109	0.104	0.274	0.643	0.235	0.134	0.084	0.226	0.439	0.249	0.129	0.088	0.258	0.557	0.222	0.142	0.111	0.257
	Vanilla (Gecko)	0.257	0.659	0.189	0.104	0.076	0.266	0.619	0.226	0.132	0.089	0.215	0.408	0.213	0.126	0.113	**0.229**	0.473	0.208	0.131	0.105	**0.242**
	GPT4o (Summarization)	0.262	0.642	0.202	0.108	0.097	0.266	0.634	0.233	0.121	0.076	0.220	0.405	0.226	0.144	0.103	0.250	0.542	0.214	0.145	0.098	0.249
	GPT4o (Task-specific)	0.249	0.634	0.188	0.097	0.078	0.262	0.621	0.221	0.125	0.080	0.220	0.418	0.209	0.142	0.110	0.253	0.542	0.220	0.140	0.110	0.246
	Gemini (Summarization)	0.262	0.629	0.194	0.113	0.114	0.260	0.624	0.213	0.129	0.074	**0.210**	0.415	0.224	0.119	0.083	0.238	0.523	0.210	0.127	0.092	0.243
	Gemini (Task-specific)	0.262	0.637	0.204	0.115	0.092	0.269	0.622	0.228	0.136	0.088	**0.210**	0.408	0.205	0.126	0.102	0.244	0.528	0.211	0.133	0.105	0.246
	Random (All)	0.317	0.729	0.248	0.163	0.130	0.352	0.752	0.290	0.213	0.152	0.280	0.481	0.265	0.193	0.182	0.279	0.566	0.251	0.155	0.143	0.307
	Random (Ticker)	0.247	0.633	0.182	0.095	0.080	0.255	0.596	0.208	0.130	0.088	0.228	0.438	0.212	0.133	0.130	0.238	0.513	0.203	0.124	0.110	**0.242**
	PEV(Mean)	0.309	0.725	0.236	0.150	0.124	0.330	0.723	0.279	0.186	0.134	0.262	0.463	0.249	0.172	0.166	0.278	0.584	0.245	0.148	0.134	0.295
	STPEV(Mean)	**0.239**	0.611	0.180	0.093	0.074	**0.253**	0.601	0.209	0.122	0.081	0.227	0.432	0.215	0.132	0.130	0.246	0.520	0.215	0.133	0.118	**0.242**
	STPEV(LR)	0.332	0.851	0.272	0.127	0.078	0.290	0.714	0.231	0.142	0.071	0.293	0.542	0.342	0.175	0.112	0.333	0.767	0.311	0.157	0.097	0.312
	STPEV(MLP)	0.820	1.263	1.280	0.394	0.345	0.565	1.031	0.605	0.431	0.195	0.397	0.446	0.275	0.557	0.311	0.490	0.625	0.526	0.469	0.341	0.568

Table 8: The overall performance on DEC across transcript-based models and different PEV and STPEV variants.

the embeddings of vanilla transcripts with those of four types of LLM fine-grained insights (two strategies and two LLMs). We sample 10 earnings from each ticker, resulting in 900 earnings, and then perform K-means clustering on 4,500 embeddings following t-SNE dimensionality reduction. Table 14 presents the clustering results. We observe that vanilla transcripts, GPT4o fine-grained, and Gemini fine-grained texts are distinctly separated from other sources, while the two strategies—summarization and task-specific—struggle to differentiate from each other. Additionally, secondary clusters emerge within the three main clusters, representing ticker-specific groupings.

E.5 Similarity Comparison

To evaluate how transcript representations correlate with other examples from the same-ticker group and the all-dataset group (comprising all 90 tickers), we compute the cosine similarity of each earnings record with both groups. This analysis is performed across different transcript representations, including two vanilla embeddings and four fine-grained LLM embeddings. As shown in Table 15, the cosine similarity for the same-ticker group is significantly higher than that for the all-dataset group, except for Random(All) representations. This finding indicates that transcripts from the same ticker are more similar than those from different tickers.

E.6 Predictions Correlation Analysis

We also calculate and compare the Pearson correlation coefficients between the predictions of various transcript-based models and those of STPEV(Mean) across different windows and quarters. From Table 10 to Table 17, We observe strong linear relationships between the predictions generated by transcript-based models and STPEV(Mean), typically beginning from the year 2021. Figure 16 further visualizes this relationship for the year 2023, comparing predictions from the Random(Ticker) model with those from STPEV(Mean). The points cluster closely around the line y = x , highlighting a strong correlation.

Year	Model	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Average
		$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	$\overline{MSE}$	MSE_3	MSE_7	MSE_{15}	MSE_{30}	
2019	DLinear	-	-	-	-	-	0.438	0.500	0.370	0.435	0.449	$\underline{0.322}$	0.567	0.284	0.258	0.178	0.235	0.456	0.220	0.147	0.117	0.332
	TSMixer	-	-	-	-	-	$\underline{0.413}$	0.535	0.244	0.278	0.595	0.330	0.554	0.299	0.392	0.076	**0.224**	0.399	0.220	0.148	0.130	**0.322**
	TimesNet	-	-	-	-	-	**0.394**	0.628	0.331	0.410	0.208	0.358	0.593	0.366	0.301	0.171	$\underline{0.230}$	0.395	0.268	0.153	0.102	$\underline{0.327}$
	FEDformer	-	-	-	-	-	0.706	0.743	0.733	0.982	0.366	0.325	0.724	0.270	0.177	0.128	0.267	0.459	0.283	0.178	0.149	0.433
	STPEV(Mean)	-	-	-	-	-	0.415	0.902	0.442	0.198	0.118	**0.299**	0.638	0.295	0.159	0.105	0.272	0.523	0.265	0.181	0.119	0.329
2020	DLinear	0.929	0.602	0.426	0.709	1.980	1.158	1.216	0.942	1.064	1.411	0.433	0.803	0.326	0.289	0.312	**0.246**	0.489	0.199	0.177	0.119	0.691
	TSMixer	0.920	0.677	0.439	0.606	1.956	$\underline{0.595}$	0.784	0.497	0.505	0.594	$\underline{0.381}$	0.784	0.301	0.229	0.211	$\underline{0.273}$	0.504	0.208	0.232	0.147	$\underline{0.542}$
	TimesNet	0.938	0.682	0.603	0.628	1.839	1.081	1.233	0.829	1.031	1.233	0.422	0.954	0.302	0.231	0.201	0.331	0.518	0.275	0.280	0.250	0.693
	FEDformer	**0.772**	0.654	0.322	0.499	1.613	0.912	1.251	0.903	0.714	0.781	0.442	0.883	0.329	0.293	0.264	0.318	0.610	0.230	0.211	0.220	0.611
	STPEV(Mean)	$\underline{0.817}$	0.725	0.477	0.695	1.370	**0.438**	0.786	0.383	0.346	0.237	**0.269**	0.685	0.196	0.112	0.083	0.311	0.536	0.275	0.281	0.151	**0.459**
2021	DLinear	0.192	0.434	0.188	0.086	0.061	**0.215**	0.422	0.190	0.124	0.123	**0.256**	0.498	0.267	0.161	0.097	0.248	0.549	0.207	0.146	0.092	**0.228**
	TSMixer	$\underline{0.191}$	0.434	0.193	0.083	0.053	0.226	0.401	0.162	0.141	0.199	$\underline{0.275}$	0.461	0.293	0.208	0.140	$\underline{0.246}$	0.542	0.207	0.149	0.087	$\underline{0.235}$
	TimesNet	0.221	0.409	0.226	0.170	0.078	$\underline{0.223}$	0.409	0.229	0.147	0.106	0.290	0.490	0.285	0.250	0.136	0.259	0.520	0.225	0.169	0.121	0.248
	FEDformer	0.254	0.463	0.235	0.175	0.141	0.267	0.472	0.233	0.187	0.177	0.305	0.553	0.292	0.203	0.173	0.272	0.540	0.277	0.175	0.095	0.274
	STPEV(Mean)	**0.156**	0.368	0.149	0.067	0.041	0.249	0.463	0.209	0.150	0.173	0.333	0.525	0.353	0.260	0.196	**0.222**	0.536	0.177	0.114	0.062	0.240
2022	DLinear	**0.244**	0.553	0.212	0.133	0.079	**0.208**	0.490	0.179	0.112	0.052	0.304	0.597	0.253	0.198	0.169	0.235	0.555	0.185	0.111	0.087	$\underline{0.248}$
	TSMixer	$\underline{0.246}$	0.560	0.214	0.121	0.090	0.229	0.484	0.196	0.157	0.080	$\underline{0.257}$	0.570	0.216	0.140	0.100	**0.229**	0.527	0.198	0.113	0.079	**0.240**
	TimesNet	0.266	0.562	0.233	0.155	0.114	0.239	0.499	0.185	0.167	0.106	0.270	0.601	0.221	0.136	0.123	$\underline{0.231}$	0.516	0.198	0.099	0.112	0.252
	FEDformer	0.282	0.627	0.281	0.142	0.078	$\underline{0.218}$	0.519	0.147	0.110	0.097	0.296	0.604	0.251	0.154	0.176	0.246	0.559	0.200	0.128	0.098	0.261
	STPEV(Mean)	0.270	0.584	0.245	0.152	0.099	0.310	0.640	0.270	0.201	0.129	**0.243**	0.592	0.219	0.104	0.057	0.278	0.599	0.236	0.183	0.095	0.275
2023	DLinear	$\underline{0.254}$	0.667	0.171	0.106	0.072	0.336	0.789	0.291	0.163	0.101	0.254	0.519	0.259	0.147	0.089	0.299	0.666	0.279	0.157	0.094	0.286
	TSMixer	0.255	0.671	0.183	0.102	0.064	$\underline{0.304}$	0.724	0.262	0.142	0.088	**0.223**	0.456	0.223	0.122	0.090	$\underline{0.265}$	0.592	0.241	0.137	0.091	$\underline{0.262}$
	TimesNet	0.268	0.652	0.189	0.117	0.116	0.365	0.838	0.325	0.169	0.128	0.296	0.599	0.302	0.177	0.107	0.341	0.741	0.322	0.182	0.119	0.318
	FEDformer	0.273	0.702	0.186	0.117	0.087	0.363	0.863	0.309	0.168	0.113	0.282	0.571	0.294	0.163	0.101	0.316	0.712	0.281	0.158	0.115	0.309
	STPEV(Mean)	**0.239**	0.611	0.180	0.093	0.074	**0.253**	0.601	0.209	0.122	0.081	$\underline{0.227}$	0.432	0.215	0.132	0.130	**0.246**	0.520	0.215	0.133	0.118	**0.242**

Table 9: The overall performance on DEC using pre-earnings volatility series.

Prompt for LLMs Direct Volatility Prediction

Company *ticker* has just released its earnings transcript. Our primary goal is to predict the volatility for the next *{prediction_window}* trading days.

Let me first clarify our target: volatility = log(std(r1, r2, ..., rn)), where ri is the return on day i in the future.
In general, higher volatility means more dramatic price fluctuations, indicating a more volatile market.

To help you understand the task, here are some previous examples of earnings call transcripts and their corresponding volatility values for *ticker* over the past 2 years (a total of 8 earnings). The most recent pair is labeled as previous 1, representing the latest past earnings, while previous 8 refers to the oldest past earnings.

Previous 1 (transcripts, volatility) pair for *ticker*: - Transcript (start): - Transcript (end). - Target (volatility for the next extprediction_window trading days): *volatility*

Previous 2 (transcripts, volatility) pair for *ticker*: - Transcript (start): - Transcript (end). - Target (volatility for the next extprediction_window trading days): *volatility*
...
...
...
Previous 8 (transcripts, volatility) pair for *ticker*: - Transcript (start): - Transcript (end). - Target (volatility for the next extprediction_window trading days): *volatility*

Now that you've reviewed the goal and examples, here's the current earnings call transcript for analysis:
- Transcript (start): *current_transcripts* - Transcript (end).

Let's proceed step by step: 1. Recognize patterns for using the transcripts of *ticker* to predict volatility.
2. Perform a comparative analysis of the current earnings transcript with the previous examples, as quarter-to-quarter performance is critical for earnings.
3. Use your identified patterns and comparative analysis to predict the volatility for the next *prediction_window* days.

Details about your reasoning process are highly appreciated.

Figure 11: Prompt template for LLMs direct volatility prediction.

Year	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Yearly Average
	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	
2019	-	-	-	-	-	0.14	0.169	0.146	0.125	0.121	0.477	0.483	0.456	0.472	0.496	0.415	0.469	0.431	0.39	0.37	0.556
2020	0.556	0.601	0.531	0.548	0.546	0.521	0.58	0.583	0.482	0.437	0.555	0.603	0.549	0.533	0.537	0.62	0.651	0.632	0.633	0.563	0.767
2021	0.767	0.755	0.801	0.766	0.745	0.8	0.788	0.807	0.812	0.794	0.801	0.803	0.789	0.831	0.781	0.845	0.782	0.87	0.862	0.866	0.87
2022	0.87	0.857	0.866	0.873	0.884	0.81	0.794	0.794	0.842	0.809	0.853	0.82	0.863	0.873	0.856	0.866	0.835	0.885	0.877	0.866	0.873
2023	0.873	0.843	0.897	0.89	0.864	0.869	0.847	0.877	0.874	0.878	0.838	0.778	0.835	0.864	0.877	0.866	0.852	0.859	0.87	0.88	0.866

Table 10: The Pearson Correlation Coefficients between vanilla(OpenAI) model and STPEV(Mean) model.

Year	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Yearly Average
	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	
2019	-	-	-	-	-	0.113	0.128	0.119	0.091	0.114	-0.003	-0.076	0.092	0.022	-0.052	0.369	0.421	0.405	0.329	0.319	0.494
2020	0.494	0.566	0.508	0.451	0.45	0.372	0.525	0.382	0.382	0.198	0.439	0.542	0.447	0.39	0.378	0.579	0.603	0.567	0.577	0.57	0.628
2021	0.628	0.616	0.659	0.618	0.62	0.707	0.706	0.702	0.717	0.704	0.737	0.733	0.731	0.753	0.729	0.732	0.719	0.741	0.744	0.724	0.753
2022	0.753	0.73	0.754	0.754	0.772	0.709	0.682	0.714	0.717	0.72	0.755	0.719	0.764	0.77	0.768	0.799	0.764	0.794	0.812	0.824	0.825
2023	0.825	0.805	0.825	0.834	0.838	0.817	0.809	0.822	0.821	0.815	0.816	0.79	0.818	0.831	0.825	0.81	0.816	0.801	0.813	0.81	0.81

Table 11: The Pearson Correlation Coefficients between vanilla(Gecko) model and STPEV(Mean) model.

Year	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Yearly Average
	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	
2019	-	-	-	-	-	0.258	0.352	0.23	0.239	0.213	0.535	0.594	0.509	0.525	0.512	0.555	0.593	0.539	0.562	0.528	0.683
2020	0.683	0.704	0.682	0.69	0.656	0.654	0.72	0.671	0.667	0.558	0.605	0.655	0.578	0.571	0.615	0.693	0.729	0.689	0.678	0.678	0.773
2021	0.773	0.775	0.779	0.76	0.778	0.752	0.782	0.762	0.741	0.725	0.843	0.879	0.842	0.835	0.817	0.863	0.848	0.889	0.865	0.852	0.799
2022	0.799	0.807	0.796	0.798	0.796	0.781	0.771	0.771	0.788	0.796	0.822	0.812	0.819	0.842	0.817	0.883	0.865	0.883	0.894	0.892	0.792
2023	0.792	0.803	0.836	0.833	0.695	0.842	0.828	0.838	0.852	0.847	0.816	0.764	0.839	0.832	0.828	0.848	0.772	0.861	0.876	0.884	0.848

Table 12: The Pearson Correlation Coefficients between GPT-4o(Summarization) model and STPEV(Mean).

Year	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Yearly Average
	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	
2019	-	-	-	-	-	0.259	0.326	0.245	0.217	0.247	0.507	0.51	0.497	0.511	0.51	0.453	0.484	0.422	0.477	0.429	0.711
2020	0.711	0.708	0.714	0.716	0.707	0.668	0.729	0.698	0.669	0.575	0.638	0.694	0.628	0.606	0.625	0.731	0.707	0.732	0.752	0.731	0.777
2021	0.777	0.792	0.788	0.753	0.777	0.86	0.86	0.855	0.871	0.854	0.833	0.839	0.82	0.829	0.844	0.821	0.831	0.839	0.816	0.798	0.831
2022	0.831	0.84	0.843	0.829	0.813	0.87	0.861	0.851	0.898	0.871	0.887	0.853	0.899	0.901	0.897	0.893	0.876	0.893	0.895	0.909	0.855
2023	0.855	0.834	0.861	0.871	0.853	0.875	0.857	0.885	0.884	0.876	0.892	0.878	0.883	0.901	0.904	0.916	0.895	0.919	0.919	0.93	0.916

Table 13: The Pearson Correlation Coefficients between GPT-4o(Task-Specific) model and STPEV(Mean).

Year	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Yearly Average
	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	
2019	-	-	-	-	-	0.27	0.409	0.262	0.211	0.197	0.598	0.639	0.6	0.579	0.575	0.507	0.522	0.469	0.53	0.509	0.726
2020	0.726	0.739	0.729	0.722	0.714	0.698	0.735	0.713	0.7	0.644	0.669	0.701	0.669	0.637	0.672	0.741	0.759	0.723	0.755	0.728	0.837
2021	0.837	0.827	0.849	0.843	0.831	0.851	0.845	0.859	0.852	0.848	0.872	0.872	0.86	0.898	0.859	0.843	0.836	0.859	0.86	0.816	0.852
2022	0.852	0.876	0.869	0.854	0.808	0.853	0.854	0.836	0.877	0.844	0.838	0.809	0.844	0.864	0.837	0.898	0.884	0.902	0.911	0.893	0.854
2023	0.854	0.844	0.859	0.863	0.851	0.866	0.845	0.857	0.867	0.895	0.845	0.84	0.825	0.85	0.864	0.867	0.858	0.843	0.88	0.888	0.867

Table 14: The Pearson Correlation Coefficients between Gemini(Summarization) model and STPEV(Mean).

Year	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Yearly Average
	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	
2019	-	-	-	-	-	0.216	0.22	0.202	0.207	0.237	0.469	0.574	0.496	0.423	0.383	0.494	0.494	0.492	0.52	0.471	0.603
2020	0.603	0.595	0.592	0.606	0.62	0.692	0.763	0.725	0.669	0.612	0.575	0.567	0.594	0.563	0.577	0.728	0.73	0.719	0.737	0.724	0.768
2021	0.768	0.78	0.799	0.743	0.749	0.785	0.805	0.816	0.781	0.739	0.789	0.83	0.772	0.786	0.769	0.791	0.771	0.808	0.804	0.783	0.816
2022	0.816	0.818	0.814	0.835	0.795	0.862	0.821	0.882	0.879	0.868	0.821	0.78	0.831	0.846	0.829	0.878	0.853	0.888	0.891	0.881	0.843
2023	0.843	0.826	0.865	0.855	0.824	0.867	0.828	0.874	0.88	0.883	0.845	0.827	0.838	0.856	0.858	0.867	0.858	0.861	0.872	0.877	0.867

Table 15: The Pearson Correlation Coefficients between Gemini(Task-Specific) model and STPEV(Mean).

Year	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Yearly Average
	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	
2019	-	-	-	-	-	0.101	0.064	0.11	0.113	0.118	0.025	0.014	0.054	0.03	0.003	-0.116	-0.018	-0.073	-0.142	-0.23	-0.162
2020	-0.162	-0.158	-0.171	-0.17	-0.149	-0.014	-0.045	-0.043	0.019	0.013	0.102	0.086	0.127	0.118	0.075	-0.124	-0.139	-0.138	-0.121	-0.097	-0.082
2021	-0.082	-0.03	-0.078	-0.109	-0.11	0.236	0.163	0.254	0.247	0.278	-0.106	-0.11	-0.082	-0.08	-0.149	0.119	0.069	0.111	0.173	0.124	0.183
2022	0.183	0.172	0.19	0.179	0.192	0.059	0.088	0.044	0.056	0.047	0.111	0.116	0.094	0.114	0.122	-0.124	-0.072	-0.125	-0.134	-0.165	0.007
2023	0.007	-0.111	0.026	0.036	0.075	0.096	0.065	0.142	0.092	0.084	-0.146	-0.091	-0.166	-0.169	-0.156	0.004	0.052	-0.019	0.006	-0.024	0.004

Table 16: The Pearson Correlation Coefficients between Random(All) model and STPEV(Mean).

Year	First Quarter					second Quarter					Third Quarter					Fourth Quarter					Yearly Average
	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	$\overline{Coef}$	$Coef_3$	$Coef_7$	$Coef_{15}$	$Coef_{30}$	
2019	-	-	-	-	-	-0.071	-0.014	-0.099	-0.068	-0.103	0.055	0.096	0.069	0.041	0.015	0.222	0.367	0.223	0.175	0.125	0.169
2020	0.169	0.253	0.154	0.138	0.133	0.316	0.499	0.358	0.219	0.189	0.308	0.379	0.342	0.273	0.238	0.506	0.606	0.541	0.481	0.395	0.47
2021	0.47	0.637	0.657	0.318	0.267	0.587	0.637	0.617	0.575	0.519	0.641	0.624	0.706	0.665	0.569	0.726	0.765	0.749	0.71	0.682	0.786
2022	0.786	0.913	0.788	0.753	0.691	0.878	0.885	0.89	0.879	0.859	0.912	0.896	0.92	0.927	0.907	0.898	0.905	0.877	0.912	0.899	0.925
2023	0.925	0.885	0.946	0.932	0.938	0.949	0.929	0.955	0.96	0.951	0.916	0.888	0.92	0.926	0.931	0.946	0.939	0.952	0.952	0.94	0.946

Table 17: The Pearson Correlation Coefficients between Random(Ticker) model and STPEV(Mean).

Prompt with *summarization* strategy

Company *ticker* released its earnings call transcripts just now. As an earnings call transcripts analyzer, here are the key aspects you need to focus on to gain meaningful insights:

1. Financial Performance Metrics

- Revenue: Compare reported revenue against analyst estimates and year-over-year (YoY) growth.
- Earnings Per Share (EPS): Assess whether EPS met, exceeded, or missed expectations.
- Profit Margins: Look at gross, operating, and net margins for any improvements or declines.
- Guidance: Check forward-looking revenue, EPS, and other projections provided by the company.

2. Management Commentary

- Tone and Language: Analyze whether management is optimistic, cautious, or defensive.
- Key Themes: Identify recurring themes or buzzwords (e.g., cost-cutting, innovation, market expansion).
- Clarity: Observe if management clearly addresses concerns or uses vague language.

3. Operational Updates

- Market Performance: Insights into geographic regions or product lines driving growth or underperformance.
- New Initiatives: Details about product launches, partnerships, acquisitions, or market expansions.
- Challenges: Discussion of supply chain issues, regulatory hurdles, or other headwinds.

...
...
...

Here is the earnings call transcripts: *current_earnings_call_transcripts*.
Earnings call transcripts ends.

Figure 12: Prompt with summarization strategy.

Prompt with *task-specific* strategy

Company *ticker* released its earnings call transcripts just now. You are going to predict the volatility following the earnings, which is defined as the log of the standard deviation of following n days returns, generally, the volatility is higher, the more dramatic the market reacts to the earnings.

Try to mine some cues from the earnings calls transcripts and clearly state why these cues can affect the volatility post earnings. More details from the cues to volatility are appreciated. Below are some possible key cues and explanations of their potential impact on volatility, you do not need to cover all aspects, just focus on those occur in transcripts.

1. Earnings Surprises

- Magnitude of earnings per share (EPS) or revenue beats/misses relative to analyst estimates.
- Large surprises (positive or negative) often lead to dramatic market reactions due to adjustments in future expectations.
- A large miss might signal fundamental issues, triggering sell-offs, while a large beat might lead to euphoria, driving prices up significantly.

2. Management Guidance

- Updates to forward-looking revenue, earnings, or margin expectations.
- Upward guidance revision increases investor confidence, often reducing downside risk but amplifying upside potential, leading to volatility.
- Downward revisions heighten uncertainty and can trigger aggressive revaluation.

3. Sentiment and Tone

- A confident tone with positive language like "record growth" or "strong demand" reduces perceived risk but might create higher expectations, increasing volatility if unmet.
- Defensive or overly cautious tone signals potential underlying issues, creating fear or speculation.

...
...
...

Here is the earnings call transcripts: *current_earnings_call_transcripts*.
Earnings call transcripts ends.

Figure 13: Prompt with *task-specific* strategy.

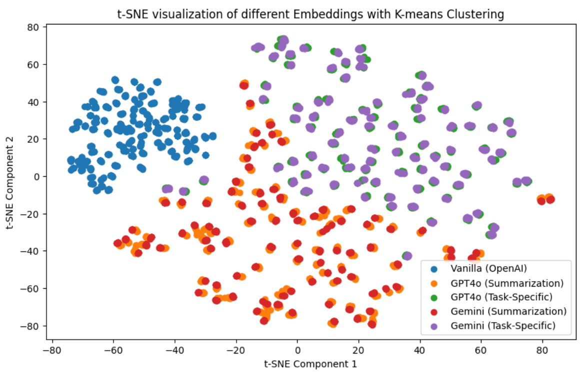
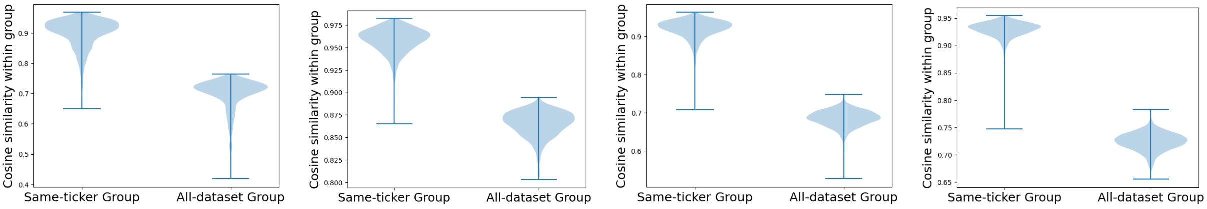
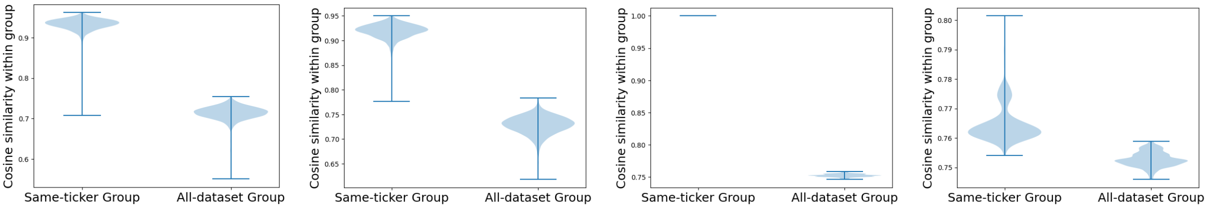


Figure 14: Clustering results for different texts.



(a) Vanilla (OpenAI) (b) Vanilla (Gecko) (c) GPT4o (Summarization) (d) GPT4o (Task-Specific)



(e) Gemini (Summarization) (f) Gemini (Task Specific) (g) Random Ticker (h) Random All

Figure 15: Cosine similarity comparison between same-ticker group and all-dataset group.

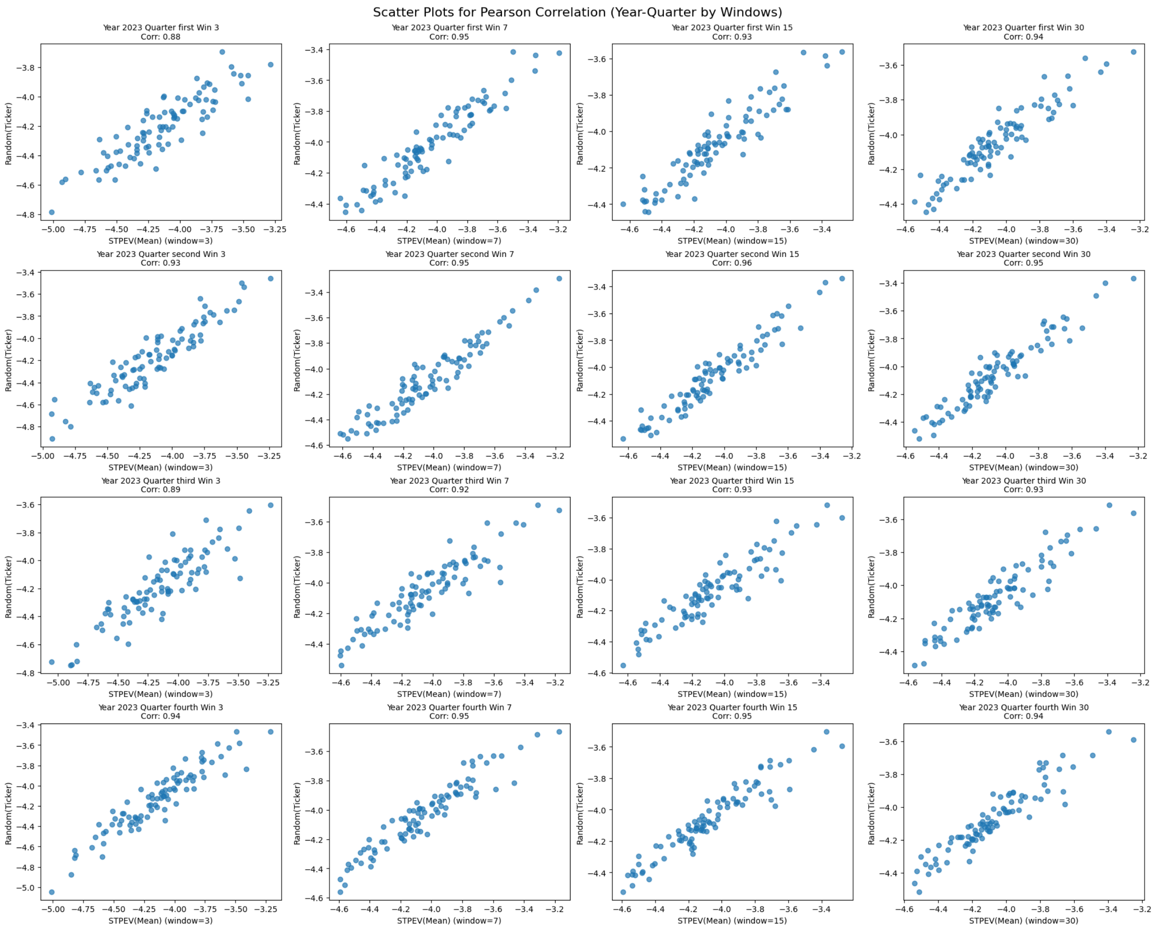


Figure 16: Predictions by Random(Ticker) - Predictions by STPEV(Mean) model for year 2023.