Assessing Consistency and Reproducibility in the Outputs of Large Language Models: Evidence Across Diverse Finance and Accounting Tasks

March 2025

Julian Junyan Wang

University College, University of Oxford

julian.wang@univ.ox.ac.uk

Victor Xiaoqi Wang

College of Business, California State University Long Beach

victor.wang@csulb.edu

Abstract: This study provides the first comprehensive assessment of consistency and reproducibility in Large Language Model (LLM) outputs in finance and accounting research. We evaluate how consistently LLMs produce outputs given identical inputs through extensive experimentation with 50 independent runs across five common tasks: classification, sentiment analysis, summarization, text generation, and prediction. Using three OpenAI models (GPT-3.5-turbo, GPT-4o-mini, and GPT-4o), we generate over 3.4 million outputs from diverse financial source texts and data, covering MD&As, FOMC statements, finance news articles, earnings call transcripts, and financial statements. Our findings reveal substantial but task-dependent consistency, with binary classification and sentiment analysis achieving near-perfect reproducibility, while complex tasks show greater variability. More advanced models do not consistently demonstrate better consistency and reproducibility, with task-specific patterns emerging. LLMs significantly outperform expert human annotators in consistency and maintain high agreement even where human experts significantly disagree. We further find that simple aggregation strategies across 3-5 runs dramatically improve consistency. We also find that aggregation may come with an additional benefit of improved accuracy for sentiment analysis when using newer models. Simulation analysis reveals that despite measurable inconsistency in LLM outputs, downstream statistical inferences remain remarkably robust. These findings address concerns about what we term "G-hacking," the selective reporting of favorable outcomes from multiple Generative AI runs, by demonstrating that such risks are relatively low for finance and accounting tasks.

Keywords: Generative AI (GenAI), Large Language Models (LLMs), ChatGPT, Reproducibility, G-hacking

Assessing Consistency and Reproducibility in the Outputs of Large Language Models: Evidence Across Diverse Finance and Accounting Tasks

1. Introduction

The integration of artificial intelligence, particularly Large Language Models (LLMs), into finance and accounting research has surged dramatically in recent years. These sophisticated models, exemplified by OpenAI's GPT series, have demonstrated remarkable capabilities in analyzing financial texts, automating complex tasks, and generating insights from vast amounts of unstructured data (Dong et al. 2024). From classifying financial text, sentiment analysis, summarization of lengthy corporate disclosures, to predicting future earnings or stock returns, LLMs promise to revolutionize how researchers approach financial text analysis (de Kok 2025) and how investors process information and make investment decisions (Blankespoor et al. 2024). The rapid adoption of LLMs in finance and accounting research is driven by several factors. First, these fields have traditionally relied heavily on textual analysis, with researchers examining corporate disclosures, analyst reports, regulatory filings, and other text-rich documents to extract insights about firm behavior, market reactions, and economic conditions. Second, the sheer volume of financial textual data has grown exponentially, making manual analysis increasingly impractical. Third, LLMs offer unprecedented capabilities for processing such data efficiently while capturing nuances that simpler computational approaches might miss. Recent studies have demonstrated LLMs' effectiveness in tasks ranging from classification (Hansen and Kazinnik 2024), sentiment analysis (Lopez-Lira and Tang 2024), summarization (A. Kim et al. 2024a), text generation (Bai et al. 2023), to prediction (Li et al. 2024).

However, as with any emerging methodological innovation, questions of consistency and reproducibility remain paramount concerns for the research community. The foundation of scientific inquiry rests on the principle that research findings should be reproducible—that independent researchers following the same methods should arrive at substantially similar results. Yet, this fundamental tenet faces a unique challenge when applied to research utilizing LLMs. Unlike deterministic algorithms with fixed outputs for given inputs, LLMs generate outputs through probabilistic token selection processes, meaning that even identical inputs with identical prompts may yield different results across multiple runs. This inherent stochasticity raises serious questions about the scientific validity and reproducibility of research findings that rely on LLM-generated outputs.

While a growing body of literature has explored LLMs' performance in finance and accounting applications (Dong et al. 2024), surprisingly little attention has been paid to this critical issue of reproducibility. Existing studies have primarily focused on comparing LLMs' performance to traditional methods, evaluating their capabilities in specific tasks, or addressing potential biases in their outputs (Levy 2024, Sarkar and Vafa 2024). The broader finance and accounting literature has grappled with reproducibility challenges even before the advent of LLMs, with researchers like Harvey (Harvey 2019) and Gow (2025) highlighting issues of p-hacking and replicability, and Menkveld et al. (2024) introducing the concept of "nonstandard errors" to characterize variability in outcomes when different researchers test identical hypotheses on the same data. However, the fundamental question of whether LLM-based analyses can produce consistent, reproducible results—a prerequisite for their integration into rigorous academic research—remains largely unexplored.

The reproducibility concerns extend beyond academic settings to practical applications in investment and financial decision-making contexts. Blankespoor et al. (2024) document the widespread adoption of Generative AI among retail investors, with nearly half of surveyed investors already using these technologies for tasks ranging from interpreting financial information to calculating ratios and assessing sentiment. For these users, inconsistent outputs could lead to conflicting interpretations of the same underlying information, creating potential risks for misinformed financial decisions.

Our study addresses this critical gap by conducting the first comprehensive assessment of the consistency and reproducibility of LLM outputs in finance and accounting research tasks. We choose three GPT models from OpenAI—GPT-3.5-turbo, GPT-4o-mini, and GPT-4o—due to the current dominance of GPT models in both research (Dong et al. 2024) and among investors (Blankespoor et al. 2024). Our assessment covers five key tasks in finance and accounting research: classification, sentiment analysis, summarization, text generation, and prediction. We employ a rigorous experimental design involving 50 independent runs for each task conducted over 50 days, using different random seeds to capture the models' inherent stochasticity. The analysis spans diverse financial text sources, including MD&As from 10-K filings, FOMC statements, financial news articles, earnings conference call transcripts, and comparative financial statements. Through this extensive approach, we generate more than 3.4 million outputs through approximately 750,000 API calls, creating one of the most comprehensive assessments of LLM reproducibility to date, based on more than 80 million pairwise comparisons across runs.

For each task, we implement a zero-shot prompting approach, focusing solely on the models' inherent capabilities without task-specific examples or fine-tuning. This methodological choice provides a more stringent test of the models' fundamental understanding and generalization capabilities while minimizing potential biases from few-shot examples or Chain-of-Thought (CoT) prompting. To assess consistency, we employ a comprehensive set of metrics tailored to each task's output type, including inter-rater agreement measures (Fleiss' Kappa, Krippendorff's Alpha), correlation metrics (intra-class, Pearson, Spearman, and Concordance), variation metrics (Mean Absolute Relative Differences), and other run-level and document-level agreement metrics, for nuanced analysis of both categorical and continuous outputs. We also assess the consistency of text outputs in semantic similarity and characteristics such as tone and length. We measure semantic similarity using cosine similarities calculated from text embeddings generated by a state-of-the-art embedding model (*jina-embeddings-v3*), and we measure tone using both traditional word lists (Loughran and McDonald 2011) and FinBERT (Huang et al. 2023).

Our findings reveal substantial—but task-dependent—consistency across models and tasks. For binary classification tasks like identifying forward-looking statements, both GPT-3.5-turbo and GPT-4o-mini achieve near-perfect consistency (Fleiss' Kappa > 0.93), with identical classifications across all 50 runs for over 92% of sentences. Sentiment analysis similarly demonstrates high consistency (Fleiss' Kappa > 0.94) across all text types, though with notable differences in sentiment distribution between models. More complex tasks show greater variability, with multi-class classification of FOMC statements exhibiting moderate consistency (Fleiss' Kappa = 0.86-0.91), while numerical predictions of future earnings display substantial variation in point estimates from less advanced models.

Importantly, we find that more advanced models do not consistently outperform their predecessors in terms of consistency across all tasks. While GPT-4o demonstrates superior consistency in numerical predictions relative to earlier models, GPT-3.5-turbo unexpectedly outperforms GPT-4o-mini in multi-class classification of FOMC statements. These findings suggest that model sophistication does not necessarily translate to increased consistency, and task-specific considerations should guide model selection.

For tasks generating textual outputs (summarization and text generation), we observe high semantic consistency (mean cosine similarity > 0.94) but moderate variation in length and tone. The semantic content of summaries and generated answers remains highly stable across runs, but word counts can vary by 8-15% on average, with substantially higher variation for some documents. These results suggest that while LLMs maintain consistent core messaging, researchers should exercise caution when analyzing structural or stylistic features of generated texts.

We conduct additional analyses addressing three important research questions. First, we investigate whether LLMs exhibit similar inconsistency patterns to expert human annotators in classification tasks. Using the Financial PhraseBank dataset (Malo et al. 2013, Yoo 2024), we compare LLM and human expert consistency in sentiment classification. Our findings reveal that LLMs demonstrate higher consistency than human annotators in over 50% of cases, with human experts outperforming LLMs in fewer than 9% of sentences. Most strikingly, LLMs maintain high agreement levels (96.8-98.3%) even for sentences where human annotators significantly disagree. This suggests that LLMs develop more deterministic classification boundaries that remain stable regardless of the inherent ambiguity in texts that causes human disagreement, making them potentially valuable tools for tasks requiring consistent classifications.

Second, we examine whether aggregation strategies across multiple runs can enhance consistency. For both classification tasks and continuous variables, we find that even modest aggregation substantially improves reliability.[1] The most dramatic improvements occur within the first 3-5 runs, with majority class strength for classification tasks increasing from approximately 86% to 94% for GPT-3.5-turbo and from 82% to 91% for GPT-4o-mini in this initial range. For

[1] For aggregating multiple runs, majority voting is used for categorical outputs, while averaging is used for continuous outputs like text length. For example, if three runs are aggregated, then the length is the average from three runs.

continuous variables like text length, the variation between model runs decreases dramatically from approximately 8.5% with a single run to about 3% with just five aggregated runs. We also find that aggregation can improve the accuracy of sentiment analysis when using newer models. These findings provide practical guidance for researchers seeking to balance computational costs with reliability requirements, as most consistency benefits can be captured with minimal additional computational overhead, possibly with the added benefit of improved accuracy.

Finally, we evaluate how LLM output inconsistency affects downstream statistical analyses. Through extensive simulation with 10,000 iterations and over 10 million regressions, we examine how variability in LLM-generated document lengths propagates into regression results. Despite measurable inconsistency in document summarization, we find that regression coefficients remain unbiased, t-statistics maintain their expected distributions, and inference validity is preserved at remarkably high rates—97.35% of all inferences drawn about the significance of length effects are correct, with Type I and Type II errors occurring at only 1.28% and 1.38% respectively. Most notably, when significant effects are detected, they almost never show an incorrect sign direction. These results provide strong reassurance that LLM-based document processing can be reliably incorporated into quantitative research frameworks, even when analyzing subtle effects where measurement precision is important.

This paper makes several important contributions to the emerging literature on LLMs in finance and accounting research. First, it provides robust empirical evidence on the degree of consistency in LLM outputs for common research tasks, offering researchers essential guidance on the reliability of these models when incorporated into their methodological toolkits. Our findings establish benchmark expectations for reproducibility across different tasks, allowing researchers to anticipate potential variability and implement appropriate validation procedures.

Second, our study compares consistency across different OpenAI model versions (GPT-3.5-turbo, GPT-4o-mini, and GPT-4o), revealing that advanced models do not necessarily demonstrate improved reproducibility alongside their enhanced capabilities. This comparison offers insights into the relationship between model sophistication and output stability, a crucial consideration for researchers weighing the costs and benefits of different model choices.

Third, we examine whether LLMs exhibit similar patterns of inconsistency as expert human annotators, particularly in areas where human judgments frequently diverge. Our analysis on the Financial PhraseBank dataset provides valuable perspective on LLMs' potential as substitutes or complements to human annotation in financial text analysis.

Fourth, our research evaluates whether aggregation strategies can effectively mitigate inconsistency issues. By assessing techniques like majority voting for classifications and averaging for continuous predictions, we provide practical guidance for researchers seeking to enhance LLM reliability with modest additional computational overhead.

Finally, we assess how output variability affects downstream analyses when LLM outputs serve as variables in regression models. Our extensive simulation demonstrates that despite measurable inconsistency in LLM-generated variables, statistical analyses remain remarkably robust, providing reassurance for researchers incorporating these tools into quantitative frameworks.

Our results have significant implications for how researchers should implement, validate, and report LLM-based analyses in finance and accounting research. They highlight the importance of conducting multiple runs and reporting consistency metrics alongside performance metrics when using LLMs for research purposes. They also demonstrate that aggregating outputs across multiple runs can substantially improve consistency for most tasks, offering a practical strategy for enhancing reproducibility.

Our findings also address emerging concerns about what we term "G-hacking" in LLM-based research—a concept we introduce in this paper to describe a practice where researchers selectively report favorable outcomes from multiple runs of a Generative LLM, analogous to p-hacking in statistical analyses. We define G-hacking as occurring when researchers generate multiple outputs from generative LLMs for the same task and cherry-pick results that support their hypotheses while discarding contradictory outcomes. Our analysis suggests the risk of G-hacking is relatively low for most finance and accounting applications, as LLMs demonstrate high consistency in classification and sentiment analysis tasks, with limited variability even in more complex applications like summarization and text generation. However, the practice remains a theoretical concern for tasks with higher variability, such as numerical predictions using less advanced models. We recommend that researchers implement transparent reporting protocols, including disclosing the number of model runs and aggregation strategies if used, to maintain research integrity. This practice, combined with our finding that simple aggregation strategies substantially improve consistency and possibly accuracy, provides a practical approach to mitigating potential G-hacking concerns while enhancing overall reproducibility.

The remainder of this paper proceeds as follows: Section 2 describes background, related studies, and our research questions. Section 3 details our experimental design, including task selection, model choice, prompting strategies, and evaluation metrics. Section 4 presents results for each task, analyzing consistency patterns and comparing performance across models. Section 5 provides additional analyses on human-LLM comparison, aggregation strategies, and downstream effects. Section 6 discusses the implications of our findings for research methodology and offers conclusions. Detailed experimental design for each task and comprehensive result interpretations are provided in Online Appendix A and Online Appendix B.

2. Background, Related Work, and Research Questions

The application of artificial intelligence (AI) and machine learning (ML) in finance and accounting research has gained significant traction in recent years (Bochkay et al. 2022). However, the use of LLMs in these fields is still at an early stage (Dong et al. 2024), particularly regarding issues of replicability and consistency. This section reviews relevant literature to contextualize our study.

2.1 LLMs in Accounting and Finance

Recent studies have demonstrated the remarkable potential of AI and ML in advancing various aspects of accounting and finance research. The emergence of sophisticated LLMs has revolutionized how researchers process and analyze financial texts, offering unprecedented capabilities for tasks ranging from classification to prediction (de Kok 2025, Dong et al. 2024). For example, Hansen and Kazinnik (2024) demonstrate LLMs' superior performance in analyzing Federal Open Market Committee (FOMC) statements. Multiple studies have established LLMs' ability to extract sentiment from financial texts with greater nuance than dictionary-based and other ML-based approaches (e.g., Bond et al. 2025, J. Chen et al. 2023, Lopez-Lira and Tang 2024).
Li et al. (2023) and Choi and Kim (2023) highlight LLMs' effectiveness in classifying ESG-related disclosures and tax audit discussions in corporate reports. Wang and Wang (2024) use LLMs to extract qualitative and quantitative data from unstructured finance text. Studies by Kim et al. (2024a, b) have shown LLMs' effectiveness in summarizing MD&As and assessing risk exposures from earnings call transcripts. Additionally, Li et al. (2024) and Kim et al. (2024) have explored LLMs' capacity to forecast financial performance, finding that these models can sometimes

outperform professional analysts in predicting future earnings. Shaffer and Wang (2024) demonstrate that large language models (LLMs) can effectively estimate core earnings from financial disclosures, outperforming traditional measures in predicting future profitability. These studies underscore the growing relevance of Generative AI in finance and accounting.

2.2 Replicability Concerns in Finance and Accounting Research

Replicability has been a longstanding concern in finance and accounting research, even before the introduction of LLMs. Harvey (2019) discussed issues of p-hacking and replicability in finance research, highlighting the importance of editorial policies for robust and reproducible results. Menkveld et al. (2024) introduce the concept of "nonstandard errors" to characterize the variability in research outcomes when different research teams test the same hypotheses. Building on this same experimental framework, Pérignon et al. (2024) specifically examine computational reproducibility, finding that only 52% of finance research results could be precisely reproduced even when research teams provided their code and documentation.

Interestingly, reproducibility success appears to vary substantially across finance subfields. While many studies paint a concerning picture, Jensen et al. (2023) offer more optimistic findings in asset pricing research. Chen and Zimmermann (2022) present relatively positive assessments of reproducibility, while Hou, Xue, and Zhang (2020) and Linnainmaa and Roberts (2018) find that numerous documented effects fail to replicate under more stringent statistical tests or disappear when tested out-of-sample.

In the accounting domain, Hail et al. (2020) survey 136 accounting researchers about reproducibility concerns. Their findings reveal that 52% of respondents view irreproducibility as a major problem, with participants estimating that only 46% of published accounting results are replicable and 59% are reproducible. Like finance researchers, accounting scholars identified selective reporting, publication pressure, and poor statistical analysis as key contributing factors to reproducibility challenges. Gow (2025) provides evidence of the prevalence of p-hacking in accounting research and suggests that encouraging replication could significantly improve empirical accounting research. In a cross-disciplinary study, Fišar et al. (2024) evaluate the impact of a premium business journal’s data and code policy, finding that reproducibility rates significantly improved after policy implementation, though data accessibility and documentation challenges remain key barriers.

These reproducibility challenges in finance and accounting take on heightened significance with the integration of LLMs into research workflows. LLMs introduce additional layers of complexity through their stochastic generation processes, potentially exacerbating existing replicability concerns. As financial research increasingly incorporates these tools, the field must develop new methodological frameworks and verification standards that account for both traditional reproducibility challenges and the novel complications introduced by Generative AI systems.

The stochastic nature of LLMs introduces a novel challenge to reproducibility and replicability that we term "G-hacking"—the practice of selectively reporting favorable outcomes from multiple model runs of a Generative AI model while discarding contradictory results. This concept extends traditional concerns about p-hacking into the domain of Generative AI research. G-hacking represents a particular risk in LLM-based studies because the inherent randomness in model outputs provides researchers with plausible deniability for selective reporting. The opacity of LLM generation processes could potentially mask selective reporting practices. This emerging concern further underscores the importance of establishing baseline expectations for LLM output consistency across different tasks and model types.

2.3 Methodological Approaches to AI Reproducibility

Several studies have proposed methods for assessing and ensuring reproducibility in AI and ML research. Raff (2019) evaluates reproducibility in machine learning by attempting to reproduce 255 papers published between 1984 and 2017. Statistical analyses identify key reproducibility factors including manuscript clarity, algorithm complexity, hyperparameter documentation, and researcher responsiveness. Expanding on these challenges, Gundersen and Kjensmo (2018) provide a comprehensive overview of reproducibility issues in AI research, offering insights that could be applied to LLM studies. They develop a taxonomy of reproducibility in empirical AI research and propose documentation standards to enhance replicability. Contributing to this methodological framework, Pineau et al. (2021) introduce a reproducibility checklist for ML research, emphasizing the importance of detailed reporting of experimental conditions, data preprocessing steps, and model specifications. Ahmed and Lofstead (2022) examine the challenge of reproducibility in machine learning, focusing on controlling randomness during model training.

2.4 Related Studies on LLM Reliability

Recent studies have explored various aspects of LLMs and their applications in finance, accounting, and many other disciplines. However, the consistency and reproducibility of LLMs' performance have not been extensively investigated.

Aguda et al. (2024) investigate the potential of LLMs as data annotators for extracting relations in financial documents. They emphasize the importance of customizing prompts for each relation group and introduce a reliability index to identify outputs that may require expert attention. This work provides valuable insights into using LLMs for structured information extraction but does not examine output consistency across multiple runs. Reiss (2023) examine the consistency of ChatGPT's zero-shot capabilities for text annotation and classification, finding that even minor wording alterations in prompts or repeating identical inputs can lead to varying outputs. The study underscores the need for thorough validation against human-annotated data and represents one of the few works directly addressing LLM consistency, though with a narrower scope than our comprehensive assessment.

In the domain of data annotation, Gilardi et al. (2023) and Törnberg (2023) demonstrate that ChatGPT outperforms crowd-workers and experts in annotating political Twitter messages with zero-shot learning, achieving higher accuracy, reliability, and equal or lower bias. These studies establish LLMs' potential to match or exceed human performance in certain annotation tasks but do not examine the consistency of these outputs across multiple runs. Exploring another dimension of LLM applications, Rossi et al. (2024) investigate the use of LLM-generated synthetic data in social science and design research, challenging some of the underlying assumptions and highlighting the main challenges that need to be addressed. Their work raises important questions about the validity of LLM-generated content for research purposes but does not systematically evaluate output consistency.

Addressing a critical concern in financial applications, Sarkar and Vafa (2024) discuss the issue of temporal lookahead bias in empirical analysis that uses outputs from pretrained language models. They develop direct tests for lookahead bias and discuss the limitations of prompting-based approaches to counteract this bias. Yoo (2024) evaluates the reliability of self-reported confidence scores from LLMs in sentiment analysis of finance news. The study reveals significant overconfidence in ChatGPT's own accuracy assessment and suggests alternative methods, such as fine-tuning, to obtain more reliable results. This work highlights important limitations in LLMs' self-assessment capabilities but does not address the fundamental question of output reproducibility. Adding to these insights, Levy (2024) aims to provide rationale for the superior performance of Generative AI systems in accounting and finance-related tasks, highlighting the poor numerical reasoning of LLMs and the presence of look-ahead bias in prediction tasks. This analysis offers important caveats regarding LLMs' capabilities but focuses on performance limitations rather than consistency issues.

While these studies have provided valuable insights into the capabilities and limitations of LLMs, they have primarily focused on aspects such as performance, overconfidence, look-ahead bias, and numerical reasoning abilities. In contrast, our study aims to fill a critical gap in the literature by specifically assessing the consistency and reproducibility of LLMs' performance across multiple runs for various natural language processing tasks in finance and accounting. By examining the consistency of LLMs' outputs, this study contributes to a more comprehensive understanding of their reliability and robustness, which is essential for their practical application in research settings where reproducibility is a fundamental requirement.

The findings of our study complement the existing research by addressing a crucial aspect that has not been extensively investigated. The results have significant implications for researchers and practitioners who rely on LLMs for tasks such as classification, sentiment analysis, text generation, summarization, and predictions, as they provide concrete insights into the reproducibility and trustworthiness of LLM-generated outputs under different conditions and across various tasks.

2.5 Practical Implications for Financial Decision-Making

This reproducibility concern extends beyond academic settings to practical applications in investment and financial decision-making contexts. Recent evidence from Blankespoor et al. (2024) documents the widespread adoption of Generative AI among retail investors, with nearly half of surveyed investors already using these technologies. Their survey data reveals that investors employ LLMs for a diverse range of financial processing tasks—44.2% ask for explanations or interpretations of financial information, 41.2% request definitions of financial terms, 28.7% search for specific mentions of companies or industries, and 28.2% retrieve financial information from the Internet. More sophisticated analytical tasks are also common, with 21.4% using generative AI to calculate financial ratios, 21.0% summarizing financial documents, 16.4% identifying trends in financial data, and 14.9% assessing sentiment. Notably, 27.4% of a subset of users specifically report asking for financial decision recommendations.

The reproducibility challenges of LLMs are particularly concerning for these specific applications. Unlike traditional quantitative models with deterministic outputs, LLMs may produce inconsistent analyses of identical financial data across different sessions or model versions. For investors relying on LLMs to explain financial concepts, identify trends, calculate ratios, or assess sentiment, inconsistent outputs could lead to conflicting interpretations of the same underlying information. This variability creates potential risks for decision-makers who rely on these tools for time-sensitive or high-stakes financial decisions, especially the more than one-quarter of users who directly seek investment advice from these systems. Even as Blankespoor et al. (2024) report that nearly two-thirds of investors plan to continue or start using Generative AI and believe it will become a standard tool for investors, many remain skeptical due to concerns about response quality—concerns that are fundamentally linked to reproducibility challenges.

Furthermore, the "black box" nature of many LLM implementations compounds these concerns, as financial practitioners may have limited visibility into how specific outputs are generated or what factors might cause variations in results. This opacity, combined with reproducibility challenges, necessitates new approaches to model governance and output verification in financial applications where decision reliability and consistency are paramount.

2.6 Research Questions

This study investigates the consistency and reproducibility of LLMs for common finance and accounting research tasks. We systematically assess output variability across multiple runs and compare performance across different model versions, while also examining whether LLMs exhibit inconsistency patterns similar to human annotators. The following five research questions guide our investigation:

RQ1: How consistent are LLM outputs for common finance and accounting research tasks across multiple runs?

This question evaluates the fundamental consistency of LLMs when provided with identical inputs and prompts across multiple executions. By subjecting the models to repeated runs on the same tasks, we aim to quantify output variability and assess their ability to produce reliable, reproducible results. This analysis is essential for establishing baseline expectations regarding LLM robustness in finance and accounting applications, where consistency is a prerequisite for scientific validity and reliable decision-making.

RQ2: How do different versions of LLMs compare in terms of output consistency for finance and accounting tasks?

This question examines whether more advanced model versions demonstrate improved consistency alongside their enhanced capabilities. By systematically comparing output reliability across different models (e.g., GPT-3.5-turbo, GPT-4o-mini, and GPT-4o), we determine whether consistency improves with model sophistication or follows different patterns. These insights are critical for researchers and practitioners weighing trade-offs between cost, performance, and reliability when selecting appropriate models for specific applications.

RQ3: Do LLMs exhibit similar inconsistency patterns to human annotators in classification tasks?

This question investigates whether LLMs mimic human annotation patterns, particularly in contexts where human annotators demonstrate known disagreements. By analyzing model outputs on datasets with documented human inconsistencies, we assess whether LLMs struggle with the same ambiguous cases or manifest distinct variability patterns. These findings have important implications for determining when LLMs might effectively complement or potentially replace human annotation in financial text analysis.

RQ4: Can aggregation strategies across multiple runs enhance the consistency of LLM outputs?

This question explores practical mitigation strategies for LLM inconsistency through output aggregation. By implementing and evaluating various aggregation techniques, including majority voting for categorical outputs and averaging for continuous predictions, we assess their effectiveness in enhancing result reliability. Additionally, we examine the effect of aggregation on accuracy. These findings provide actionable guidance for researchers and practitioners seeking to maximize consistency in LLM-based analyses while balancing computational costs.

RQ5: How does LLM output inconsistency affect downstream statistical analyses and what are the implications for "G-hacking"?

This question examines both the propagation of LLM inconsistency into subsequent analytical workflows and addresses concerns about selective reporting of favorable LLM outputs—a practice we term "G-hacking." By assessing how variability in model outputs influences statistical significance, effect sizes, and substantive conclusions when these outputs serve as variables in regression analyses, we quantify the potential risks of incorporating LLM-generated variables into broader research frameworks. This analysis also allows us to evaluate the severity of G-hacking concerns in accounting and finance applications, providing guidance for transparent research practices that maintain analytical integrity when leveraging LLMs in empirical research.

These research questions are interconnected and collectively enable us to provide a comprehensive assessment of consistency and reproducibility in Large Language Model outputs in finance and accounting contexts.

3. Experimental Design

In this section, we describe our experimental design, covering task selection, model choice, prompting strategies, and evaluation metrics. Specific design choices for each task, such as sample selection, prompts, and other implementation details, are discussed in Online Appendix A.

3.1 Task Selection

Our task selection is guided by a comprehensive literature review conducted by Dong et al. (2024), who find that numerous studies have evaluated the capabilities of LLMs in performing various tasks or utilized LLMs to carry out tasks such as classification, sentiment analysis, summarization, text generation, and prediction. We choose these tasks for our study. To assess the consistency and reproducibility of these tasks, we use a variety of text commonly employed in accounting and finance research, including FOMC statements, financial news articles, earnings conference call transcripts, and MD&As from annual reports. For the prediction task, we provide the models with condensed comparative financial statements in addition to MD&A summaries.

Table 1 summarizes these tasks as well as their data sources and sample sizes. For classification tasks, the source data includes sentences from MD&As of 10-Ks and sentences from FOMC statements, with sample sizes being 10,000 and 1,096. Sentiment analysis tasks involve news articles, prepared presentations and Q&A sections from earnings call transcripts, and MD&As from 10-Ks, with sample sizes ranging from 2,000 to 10,000 paragraphs or sentences.

The summarization task uses MD&As from 10-Ks and presentation sections from earnings calls transcripts, with sample sizes of both 1,000. The text generation task involves answering 1,000 questions from Q&A sections of earnings call transcripts. The prediction task requires the models to predict the future earnings of 1,000 firm-years based on MD&A summaries and comparative balance sheets and income statements.

In the following sub-sections, we discuss related literature and provide additional details about each task.

3.1.1 Classification

Classification involves categorizing input data into predefined classes or categories based on their features or attributes. Recent studies have shown the potential of ChatGPT and other LLMs in classifying textual data for accounting and finance research, such as identifying ESG-related disclosures (Li et al. 2023), classifying management presentations into "facts" or "opinions" (Kuroki et al. 2023), determining the nature of discussions during earnings calls (Jia et al. 2024), and assessing firms' characteristics (Bernard et al. 2023, Dasgupta et al. 2023). Choi and Kim (2023) utilize GPT-4 to identify disclosures related to tax audits in annual reports, and Meng Wang (2023) employs a GPT-based model to classify statements made by mutual fund managers in shareholder reports.

Depending on the number of labels, classification involves binary classification (two labels) and multi-class classification (three or more labels). For binary classification, we focus on a common binary classification task in accounting and finance: identifying forward-looking statements from MD&As in annual or quarterly reports. MD&As often contain forward-looking statements that help investors and other users of financial information predict a company's future earnings and cash flows. Prior research has explored various methods for identifying forward-looking statements at the sentence level.

The traditional approach involves using a dictionary of future-oriented words and phrases (e.g., Li 2010). Researchers have iteratively refined these dictionaries over time. For instance, Muslu et al. (2014) incorporate bigrams, verb conjugations, and future date references, while Bozanic et al. (2018) remove phrases associated with cautionary statements. More recently, researchers have applied supervised machine learning to this task. Brown et al. (2024) train a convolutional neural network on a dataset of manually labeled sentences.

For multi-class classification, we choose statements from the Federal Open Market Committee (FOMC) as the source text. The stock market monitors FOMC statements closely to gauge the Federal Reserve's monetary policy stance. An existing study by Hansen and Kazinnik (2024) has demonstrated the superiority of LLMs in analyzing FOMC statements.

3.1.2 Sentiment Analysis

Sentiment analysis, fundamentally a classification task, involves identifying and quantifying the emotional tone conveyed in a piece of text. This computational technique typically classifies the sentiment expressed in a given text as positive, negative, or neutral, with some approaches further differentiating the intensity of the sentiment using numeric values. Sentiment analysis can be performed using dictionary-based methods, such as the widely-used word list developed by Loughran and McDonald (2011) or through ML approaches, like BERT and its finance-specific variant, FinBERT (Huang et al. 2023).

Numerous studies have explored the application of ChatGPT and other LLMs for sentiment analysis in various financial contexts. One group of studies focuses on using LLM-based sentiment analysis to forecast stock market trends and inform trading strategies (Álvarez-Díez et al. 2024, Bond et al. 2025, J. Chen et al. 2023, B. Chen et al. 2023, Kirtac and Germano 2024, Lopez-Lira and Tang 2024). Another group of studies applies LLMs to extract sentiment from diverse financial texts, such as central bank communications, corporate filings, and cryptocurrency news, demonstrating the highly adaptability of LLMs (Breitung et al. 2023, Alonso-Robisco and Carbó 2023, Smales 2023, Bae et al. 2024).

Comparative analyses reveal that advanced LLMs consistently outperform traditional sentiment analysis methods in capturing nuanced sentiments and extracting financially relevant information (Bond et al. 2025, J. Chen et al. 2023, Hu et al. 2023, Kirtac and Germano 2024, Lopez-Lira and Tang 2024). However, studies also highlight potential biases and limitations in LLM-based sentiment analysis, emphasizing the importance of careful implementation and domain-specific adaptations (Leippold 2023, Hu et al. 2023, Glasserman and Lin 2023).

For this common task, we use multiple text sources, as prior studies have done. The first source is sentences from the Financial PhraseBank (Malo et al. 2013), which consists of sentences from financial news articles. The benefit of this dataset is that it not only has human labels but also provides information about disagreements among expert human raters. This information allows us to provide an additional analysis as to whether LLMs are more varied in sentences where humans have greater disagreements. We also include sentences from MD&As and paragraphs from earnings conference call transcripts, covering both prepared presentation and Q&A sessions, to assess the consistency of LLMs in processing and analyzing various types of financial text.

3.1.3 Summarization

Summarization involves generating a concise and coherent synopsis of a longer text while preserving its key information. Most advanced LLMs can now create reliable summaries that capture the depth and tone of the original text. Kim, Muhn, and Nikolaev (A. Kim et al. 2024a, b) use GPT-3.5-turbo to summarize MD&As from 10-Ks, conference call transcripts, and assess a company's exposure to various risks based on earnings conference call disclosures. For this task, we ask the models to summarize MD&As and earnings call presentations.

3.1.4 Text Generation

Text generation involves creating new text based on a given prompt or context, often aiming to mimic human-like writing or response. Bai et al. (2023) use LLMs to quantify the extent of new information provided by executives during earnings call Q&A sessions, while Wu, Dong, Li, and Shi (2023) employ ChatGPT to generate analyses of textual loan assessments. For this task, we ask the model to answer questions posed by financial analysts at earnings conferences as if they were the executives of the company.

3.1.5 Prediction

Prediction involves forecasting future events or values, such as stock price movements or a company's future earnings, based on historical data and patterns. Li, Tu, and Zhou (2023) find that earnings forecasts generated by GPT-4 exhibit greater forecast errors than analyst consensus, but its performance improves for firms with better information environments or higher-quality disclosures. Comlekci et al. (2023) use ChatGPT to forecast future financial performance and dividends of public companies. Kim et al. (2024) investigate the ability of GPT-4 to perform financial statement analysis similar to a professional human analyst.

3.1.6 Model Choice

In this study, we focus on GPT models provided by OpenAI, specifically GPT-3.5-turbo, GPT-4o-mini, and GPT-4o. This choice is informed by Dong et al.'s (2024) finding that most existing studies on LLMs in accounting and finance use these models, as well as OpenAI's dominant market share among retail investors (Blankespoor et al. 2024) . We also consider cost-effectiveness in our model selection. GPT-3.5 and GPT-4o-mini are relatively less expensive, while GPT-4o is more capable but also significantly more costly (more than 15 times as expensive), as shown in Panel A of Table 2.

For our classification and sentiment analysis tasks, which are considered to have low difficulty, we use the capable and cost-effective GPT-3.5-turbo and GPT-4o-mini models. For summarization, we use only GPT-4o-mini, which has a large context window and is also cost-effective. For text generation, we use both GPT-3.5-turbo and GPT-4o-mini. Our prediction task, considered to have the highest complexity due to its requirement for mathematical calculations and logical reasoning, is performed using GPT-4o, GPT-4o-mini, and GPT-3.5-turbo, allowing us to compare the performance of these models on a challenging task.

For most tasks, we employ more than one model. This choice allows us to compare the consistency across models and determine whether a later generation or more advanced model exhibits greater consistency. Table 2 presents the key characteristics and pricing information for the selected models, and the models chosen for each task.

3.1.7 Prompting Strategies and Parameter Setting

This study employs zero-shot prompting across all tasks to evaluate the inherent capabilities of LLMs without task-specific examples or fine-tuning. This methodological choice is motivated by several considerations. First, zero-shot prompting provides a more stringent test of the model's fundamental understanding and generalization capabilities, as it requires the model to perform tasks solely based on instructions without exemplars. Second, this approach minimizes potential biases that might be introduced through few-shot examples or Chain-of-thought (CoT) prompting, which could inadvertently guide the model toward specific patterns or responses. Third, zero-shot prompting offers greater operational consistency across different tasks, facilitating more direct comparisons of model performance across diverse applications. For the parameters, we set the temperature to zero for maximum consistency, a choice commonly made by most researchers, while keeping other parameters at their defaults.

3.1.8 Experimental Runs and Experiment Period

We choose to conduct 50 runs for each task to balance cost, statistical power, and the inherent randomness of LLMs. This choice aligns with our sample size selection for each run, detailed in the task-specific design provided in Online Appendix A. By incorporating large sample sizes across numerous runs, we capture both within-run and across-run variation, strengthening the generalizability of our findings. This approach is crucial given LLMs' sequential token generation process, where each token is influenced by all preceding tokens, potentially creating greater output variation even with identical inputs.

These experiments span 50 days between December 27, 2024, and February 15, 2025, conducted at varying times to ensure diverse conditions. Each run utilizes a different seed to maintain independence between runs. We intentionally avoid using identical seeds across runs, as our goal is not to assess reproducibility given the same seed, but rather to evaluate consistency under varying conditions and the inherent randomness fundamental to Generative AI models.

3.1.9 Evaluation Metrics

We evaluate consistency using a comprehensive set of metrics tailored to each task's output type: categorical variables, continuous variables, and texts. For texts, we examine the consistency of characteristics such as similarity, length, and tone, where similarity and length are continuous variables and tone is categorical.

As shown in Panel A of Table 3, we use a wide variety of metrics covering overall inter-rater agreements, run-level agreements, and document-level agreements. The overall inter-rater agreement metrics provide an overall summary metric of consistency, and they include Fleiss' Kappa (Fleiss 1971) and Krippendorff's Alpha (Krippendorff 2011). These metrics are particularly valuable for quantifying agreement across multiple runs simultaneously, with higher values indicating stronger agreement beyond what would be expected by chance.

The run-level agreements provide summary metrics at the run-level, offering insights into pairwise consistency between different model runs. These include the run-pair Cohen's Kappa score and run-pair agreement (%). The run-pair Cohen's Kappa adjusts for agreement that might occur by chance, making it a more stringent measure than simple percentage agreement. Its value ranges from -1 to 1, with higher values indicating stronger agreement beyond chance. The run-pair agreement percentage directly quantifies the proportion of documents classified identically between runs, providing an intuitive measure of classification consistency.

At the document-level, we use percentage of perfect agreement (%), document-wise agreement (%), majority class strength (%), and classification uncertainty (%). These metrics shift the focus from run-pairs to individual documents, revealing how consistently each document is classified across all runs. The percentage of perfect agreement identifies documents with complete classification consensus across all 50 runs. Document-wise agreement (%) measures, for each document, what percentage of run-pairs produce identical classifications. Majority class strength quantifies the dominance of the most frequent classification for each document, with higher percentages indicating stronger consensus. Classification uncertainty, calculated using entropy, measures the variability in classifications across runs for documents where disagreement exists, with higher values signaling greater disagreement and potentially more challenging documents.

Appendix A provides an illustration of how the run-level and document-level agreement metrics are calculated. Out of 1,225 possible run-pairs from 50 runs, there is a distribution of 1,225 run-pair agreement metrics. We provide aggregated statistics like the mean, median, and standard deviation of the distribution to characterize central tendency and variability. For the document-level metrics, their number depends on the number of documents, which is 10,000, for example, for the classification task of forward-looking statements in MD&As. We also provide descriptive statistics of their mean, median, and standard deviation, among others, for all these 10,000 documents. Please see Appendix B for the detailed definition of each metric.

For continuous outputs, as shown in Panel B of Table 3, we employ metrics specifically designed to assess consistency in numerical values across different runs. These metrics fall into four main categories: overall inter-run reliability, correlation metrics, run-level variation, and document-level variation.

The overall inter-run reliability is measured using the Intraclass Correlation Coefficient (ICC2, for type 2), which quantifies the degree of consistency among multiple runs. This metric is particularly valuable for assessing absolute agreement among runs rather than just relative consistency, with values closer to 1 indicating strong reliability. ICC2 specifically accounts for both systematic and random differences between runs, making it suitable for scenarios where we want to determine if different runs can be used interchangeably.

For other correlation metrics, we utilize Concordance, Pearson, and Spearman correlations. Concordance correlation extends beyond traditional correlation by measuring not only the linear association between run pairs but also how closely the data points follow the line of perfect agreement. This helps detect systematic deviations between runs even when they maintain strong correlations. Pearson correlations assess linear correlations between different runs. Spearman correlation focuses specifically on rank-order consistency between runs, making it robust against outliers and non-linear relationships. Together, these correlation metrics provide complementary insights into how consistently runs maintain both absolute values and relative rankings.

We further assess run-level variation using run-pair MARD (%), which represents the Mean Absolute Relative Difference between pairs of runs. This metric is particularly useful for understanding the typical magnitude of differences between runs as a proportion of their average value, with lower percentages indicating smaller relative variations.

At the document-level, we measure documents with identical output (%) and document-wise MARD (%). The percentage of documents with identical outputs directly quantifies how often runs produce exactly the same numerical value (e.g., length or amount) for a document, highlighting the proportion of cases where there is perfect agreement. Document-wise MARD provides a more nuanced view by calculating, for each document, the average relative difference across all run pairs. This helps identify documents that consistently produce higher variability across runs, potentially indicating challenging cases where the model struggles to produce stable outputs. For both the run-level and document-level metrics, we also provide descriptive statistics for all pairs of runs and all documents.

For summarization and text generation tasks, the model outputs are texts. To evaluate the consistency of these outputs, we examine several common textual attributes frequently used in accounting and finance textual analysis. These characteristics include length (word counts), tone (positive, negative, or neutral), and semantic similarity (cosine similarity), as shown in Panel C of Table 3. Since length and semantic similarity are continuous variables, their results are evaluated using the metrics for continuous variables.

To determine the tone of generated texts, we use both a traditional lexicon-based approach based on word lists from Loughran and McDonald (2011) and a more recent ML-based approach.

The latter is based on FinBERT, which is a BERT model fine-tuned on financial texts and has demonstrated superior performance over dictionary-based approaches (Huang et al. 2023). For each text, we classify its tone using both methods and evaluate the result using our chosen metrics for categorical variables.

To evaluate semantic similarity between generated texts across different runs, we utilize a state-of-the-art embedding model, *jina-embeddings-v3*, which offers an optimal balance between computational efficiency and performance (Sturua et al. 2024). The process involves converting each generated text into high-dimensional embeddings and computing pairwise cosine similarities between runs. This approach allows us to quantify how consistently the model maintains core semantic content across multiple generations. We assess the consistency of the similarity scores using our chosen metrics for continuous variables.

4. Experimental Results

In this section, we discuss the results for consistency of model outputs across multiple runs using the evaluation metrics detailed in Section 3.1.9. As shown in Table A1 in Online Appendix A, our analysis covers more than 3.4 million outputs generated through approximately 750,000 API calls. For each task, we conduct 50 independent runs, yielding 1,225 run pairs per document[2]. This results in over 84 million pairwise comparisons, providing a comprehensive basis for evaluating output consistency.

As explained in Section 3.1.9, we distinguish between agreement measures at run-pair and document levels. Run-pair agreement metrics assesses the consistency between pairs of runs across all documents, revealing how similar any two runs are in their overall classification patterns. In

[2] For each document, we generate (50 × 49)/2 = 1,225 unique run pairs. With 3.44 million total outputs (documents × 50 runs), this yields more than 84 million pairwise comparisons (3.44 million × 49/2).

contrast, document-wise agreement focuses on how consistently each specific document is classified across multiple runs, offering granular insights into classification stability at the individual document level. This helps identify the number of documents that presents classification challenges due to inherent ambiguity or potential model weaknesses. Appendix A illustrates how these two agreement measures are constructed.

We provide a summary of the key results and findings. For a more detailed discussion of the results for each task, please see Online Appendix B.

4.1 Key Findings

Our analysis reveals remarkably high consistency across most financial text analysis tasks, though with notable variations by task complexity and model. For classification tasks, both GPT-3.5-turbo and GPT-4o-mini demonstrate exceptional reproducibility, with GPT-4o-mini achieving Fleiss' Kappa values of 0.97 for binary classification and 0.86 for multi-class classification, as shown in Table 4. Interestingly, for the multi-class FOMC task, GPT-3.5-turbo slightly outperforms GPT-4o-mini with a Kappa of 0.91. Perfect agreement rates (identical classification across all 50 runs) reach 96.18% for binary tasks with GPT-4o-mini, though they drop to 55.66% for the more complex five-class FOMC statement classification. Krippendorff's Alpha scores and mean run-pair Cohen's Kappa scores are identical to Fleiss' Kappa scores when rounded to two decimal places, with minimal differences at higher precision levels. The mean classification uncertainty, which is the entropy for the documents where not all runs agree, is small, ranging from 0.34 to 0.40.

These values indicate "almost perfect" agreement according to multiple established scales. Krippendorff (2004) suggests Alpha values above 0.80 indicate "substantial" reliability. Similarly, McHugh (2012) classifies Cohen's Kappa scores of 0.80-0.90 as "strong" and above 0.9 as "almost perfect" agreement. These interpretations are widely accepted in computational linguistics studies that assess inter-rater reliability in classification tasks (e.g., Artstein and Poesio 2008, Reiss 2023). Fleiss' Kappa is simply an adapted version of Cohen's Kappa designed for more than two raters.

It is worth noting that the mean run-pair agreement (%) and mean document-wise agreement share identical mean values. This equivalence is not coincidental but by design, as both agreement metrics are derived from the same underlying pairwise comparisons and, at the highest aggregation level, converge to equal values. We present both metrics for convenience of interpretation. However, their distributions do differ. For their detailed distribution, please see relevant tables in Online Appendix B.

As shown in Table 5, sentiment analysis demonstrates consistently high reliability across all text types (news articles, MD&As, earnings call presentations, and Q&As), with GPT-4o-mini achieving mean run-pair agreement exceeding 97.5% across all document categories. All commonly used inter-rater agreement metrics such as Krippendorff's Alpha and Kappas have values well above 0.9, indicating near-perfect agreement across different runs. Mean majority class strength percentages consistently approach maximum values, while mean classification uncertainty metrics remain exceptionally low, suggesting that even in the rare instances where different runs produce disagreements, these discrepancies are minimal and do not substantially affect the overall confidence in the sentiment classifications. This remarkable consistency across multiple reliability measures reinforces the robustness of LLM-based sentiment analysis for financial text applications.

For generative tasks like summarization, semantic similarity remains consistently high (mean cosine similarity of 0.98) across both MD&As and earnings call presentations, as shown in Table 6. However, summary length shows moderate variability with Mean Absolute Relative

Differences (MARD) of approximately 8.5% between runs. Text generation tasks demonstrate strong semantic consistency (0.97 for GPT-4o-mini vs. 0.94 for GPT-3.5-turbo), though with greater length variability (8.18% vs. 14.91% MARD), as shown in Table 7. Tone consistency shows interesting patterns, with GPT-4o-mini maintaining moderate consistency (Fleiss' Kappa of 0.68 for MD&A summaries and 0.66 for text generation when measured by FinBERT). Earnings call presentations exhibit substantially higher tone consistency (run-pair agreement of 95.55%) compared to MD&As (84.40%), likely reflecting that presentations at earnings conference calls tend to be universally positive. In Online Appendix B, we provide the agreement metrics for tone measured using the finance word list (Loughran and McDonald 2011). These metrics exhibit lower agreements than those from FinBERT, consistent with findings that FinBERT better captures sentiment as demonstrated by multiple studies (Huang et al. 2023).

In earnings prediction tasks, all models maintain high directional consistency (Fleiss' Kappa ≥ 0.97), but point estimate reliability varies significantly. GPT-4o achieves near-perfect consistency with run-pair MARD of just 2.05%, substantially outperforming GPT-4o-mini (6.18%) and GPT-3.5-turbo (9.89%). GPT-4o also demonstrates superior consistency across all metrics, including ICC2 (0.997), concordance correlation (0.995), and percentage of documents with identical point estimates (29.20%). Despite the variation in other metrics, average Spearman correlations remain close to one between different runs for all three models, though GPT-4o-mini and GPT-3.5-turbo show lower values in the mid-70s to high-80s range for other correlation measures.

4.2 Comparison Across Model Versions

To address our second research question regarding inter-model consistency differences, we provide a more in-depth analysis and discussion of the results in this sub-section. Overall, we find that the relationship between model sophistication and output consistency follows task-dependent patterns rather than a universal trend. This has important implications for model selection in financial text analysis applications.

For most classification tasks, GPT-4o-mini demonstrates superior consistency compared to GPT-3.5-turbo, with statistically significant improvements in binary classification (Fleiss' Kappa of 0.97 vs. 0.93) and sentiment analysis across all document types. However, this pattern reverses for multi-class classification of FOMC statements, where GPT-3.5-turbo outperforms GPT-4o-mini (Kappa: 0.91 vs. 0.86). This suggests that newer models may not always provide more consistent outputs, particularly for tasks involving nuanced categorical distinctions.

For text generation tasks, we observe a clearer correlation between model advancement and consistency. GPT-4o-mini generates responses with significantly higher semantic similarity (0.97 vs. 0.94) and lower length variability (MARD: 8.18% vs. 14.91%) compared to GPT-3.5-turbo. The minimum document-level similarity shows particularly striking differences (0.76 vs. 0.46, as shown in Table B7 in Online Appendix B), indicating that GPT-4o-mini avoids the extreme inconsistencies occasionally produced by GPT-3.5-turbo.

The most pronounced model-dependent consistency differences appear in numerical prediction tasks. For earnings point estimates, consistency improves dramatically with model sophistication: GPT-4o achieves near-perfect reliability (MARD: 2.05%) compared to GPT-4o-mini (6.18%) and GPT-3.5-turbo (9.89%). This stepwise improvement is visually evident in the non-overlapping error distributions, with GPT-4o showing remarkably narrow dispersion centered around 2%, as shown in Panel B of Figure B6 in Online Appendix B. Please see this appendix for more details.

These findings demonstrate that while newer models generally show superior consistency, performance gaps are most pronounced in complex tasks requiring numerical precision or nuanced classification. Moreover, we observe that model outputs can exhibit systematic differences in classification distributions, even though output accuracy is not the focus of this study. For instance, GPT-4o-mini consistently identifies more forward-looking statements (24.30% vs. 17.76%) and classifies more content as positive across all text types compared to GPT-3.5-turbo. These systematic differences, rather than just random variation, suggest fundamentally different internal representations or decision boundaries between models, which researchers should carefully consider when interpreting results.

Overall, our results indicate that while newer models generally offer improved consistency, this advantage is not uniform across all tasks. The selection of appropriate models should therefore consider both the specific task requirements and the relative importance of consistency versus other factors such as cost and latency. For applications requiring precise numerical outputs or complex text generation, more advanced models provide substantial consistency benefits that may justify their higher computational demands.

5. Additional Analyses

In this section, we conduct three additional tests to address our final research questions (RQ3–RQ5). These questions examine whether LLMs exhibit similar inconsistency patterns to expert human annotators (RQ3), whether aggregation strategies can enhance LLM output consistency (RQ4), and how LLM inconsistency affects downstream statistical analyses (RQ5). To explore these issues, we analyze LLM-generated annotations on datasets with documented human disagreements, assess the effectiveness of aggregation techniques such as majority voting and averaging, and evaluate the impact of LLM variability on statistical significance and effect sizes in regression analyses. These tests provide deeper insights into the reproducibility and reliability of LLM-based methods in financial and accounting research.

5.1 GPT Models vs Human Experts

In this sub-section, we report the results for an additional analysis to answer our third research question (RQ3) regarding whether LLMs outperform expert human annotators in classification tasks in consistency and whether they exhibit similar patterns in cases where human annotators tend to disagree. For this analysis, we leverage our sentiment analysis task that utilizes the Financial PhraseBank dataset, which consists of more than 4,000 sentences with human labels and human agreement levels. The dataset comes with four levels of agreements, namely, 50%, 66%, 75%, and 100%, where 50% means 50% of human annotators agree on a label, e.g., a positive sentiment. We measure the level of agreement in the same manner whereby we calculate the agreement of the majority class for each document and we then match the results based on document identifiers.

As shown in Panel A of Figure 1, the pie charts illustrate that in 51.8% of sentences, GPT-3.5-turbo demonstrates greater consistency than human annotators. Human annotators outperform GPT-3.5-turbo in only 8.4% of cases, while ties occur in 39.8% of sentences. Similarly, GPT-4o-mini shows even stronger performance, being more consistent than humans in 52.5% of sentences, while humans outperform the model in just 4.8% of cases, with ties accounting for 42.7% of the dataset. These results indicate that both LLMs, particularly GPT-4o-mini, achieve higher consistency than human annotators in the majority of cases.

Next, we assess whether LLMs tend to also struggle with sentences where humans do. We plot the results in Panel B of Figure 1. The left plot provides the distribution of human agreements:

46.7% of sentences have perfect agreement among human annotators. However, more than 50% of sentences have varying levels of disagreement, with 24.6% showing 75% agreement, 15.8% showing 66% agreement, and 13.0% showing only 50% agreement (minimal majority consensus).

The right panel plots the average agreement levels for sentences grouped by their human agreement scores of 50%, 66%, 75%, and 100%, respectively. We find that for the 100% human agreement group, the two models exhibit agreement levels of 98.1% and 98.8% respectively, approaching perfect consistency. However, for human agreement levels of 50%, 66%, and 75%, the models maintain surprisingly high agreement levels ranging from 96.8% to 98.3%, with no significant decrease in agreement as human agreement levels decline. This suggests that LLMs maintain consistent predictions even for sentences that human annotators find ambiguous or challenging to classify. Unlike human experts, the models appear to develop more deterministic classification boundaries that remain stable regardless of the inherent ambiguity in the text that causes human disagreement.

5.2 Aggregation Across Multiple Runs to Enhance Consistency

In this sub-section, we conduct additional analyses to address our fourth research question (RQ4) regarding whether aggregating outputs from multiple runs of the same model may increase consistency. We begin by examining this question in the context of classification tasks. Specifically, we focus on the multi-class classification of FOMC statements, as distinguishing among five distinct classes presents a more challenging scenario for evaluating model consistency and reliability.

We generate our sample by keeping all sentences where not all runs agree on the label. For each document, we then randomly select a number of outputs from the 50 runs and aggregate them. The number ranges from 2 to 20. For example, at a number of 3, we aggregate the results from three runs by taking the majority class. When there is a tie, we break the tie by taking the median rank of the class, where each class is coded as 0 for 'Dovish', 1 for 'Mostly Dovish', 2 for 'Neutral', 3 for 'Mostly Hawkish', and 4 for 'Hawkish', and round up in case the median is not a whole number. We repeat this process for 50 times to generate 50 synthetic runs, each of which is the aggregation of a randomly selected three runs. We finally calculate the document-level metrics including mean majority class strength, mean classification uncertainty, mean document-wise agreement, and percentage of perfect agreement. We focus on document-level metrics rather than run-level metrics because this analysis specifically targets documents where models exhibit disagreement, allowing us to evaluate whether our aggregation approach effectively resolves classification inconsistencies in these challenging cases.

We plot the results in Figure 2. In Panel A, we observe a clear positive relationship between the number of aggregated runs and the mean majority class strength for both models, with GPT-3.5-turbo consistently outperforming GPT-4o-mini. Notably, the most dramatic improvements occur with minimal aggregation. An aggregation over 3-5 runs yields the steepest gains, with GPT-3.5-turbo improving from approximately 86% to 94% and GPT-4o-mini from 82% to 91% in this initial range. As the number of aggregated runs increases further to 20, the majority class strength for GPT-3.5-turbo rises to 98%, while GPT-4o-mini reaches about 95.5%.

The right graph in Panel A demonstrates that classification uncertainty decreases substantially as more runs are aggregated, with the entropy measure dropping most precipitously at an aggregation level of the first few runs. GPT-3.5-turbo exhibits lower uncertainty throughout the range compared to GPT-4o-mini, decreasing from around 0.34 to 0.05, indicating significantly more confident predictions with increased aggregation.

In Panel B, we see strong evidence that aggregation enhances consistency across both models. The percentage of perfect agreement shows its steepest increase at the aggregation of a few runs, with GPT-3.5-turbo jumping from approximately 25% to 65% and GPT-4o-mini from near 0% to 55% at an aggregation level of five runs. At an aggregation level of 20 runs, GPT-3.5-turbo reaches nearly 85% perfect agreement, compared to approximately 78% for GPT-4o-mini. Similarly, mean document-wise agreement scores improve steadily with additional aggregated runs, with GPT-3.5-turbo achieving over 96% agreement at 20 runs while GPT-4o-mini reaches about 94%.

These findings suggest that practitioners seeking to enhance LLM reliability need not necessarily implement extensive aggregation protocols. Even modest aggregation of just 3-5 runs can capture a substantial portion of the consistency benefits while minimizing additional computational costs. This insight is particularly valuable for applications where efficiency must be balanced against reliability requirements, as the most significant gains in consistency are realized with relatively few aggregated runs.

We next assess whether aggregation across runs also helps increase the consistency in the length of MD&A summaries. We similarly generate synthetic runs by randomly sampling and aggregating outputs, ranging from 2 to 20 runs. Since summary length is a continuous variable, we take the average of the sampled runs rather than using majority voting.

As shown in Figure 3, the benefits of run aggregation are evident in both correlation metrics and mean run-pair differences. In the left panel, all correlation metrics (ICC2, CCC, Spearman, and Pearson) demonstrate substantial improvement as the number of aggregated runs increases. The most dramatic gains occur within the first 5 aggregated runs, particularly for CCC, which starts at approximately 0.43 with a single run and increases sharply to about 0.8 with just 5 aggregated runs. The other metrics show similar patterns but with higher initial values, starting around 0.65-0.7 and improving to approximately 0.9 by an aggregation level of five runs.

As aggregation increases to 20 runs, all correlation metrics converge toward values between 0.95 and 0.98, indicating extremely high consistency. The right panel shows the Mean Run-pair MARD decreasing dramatically from approximately 8.5% with a single run to about 3% with just 5 aggregated runs, eventually reaching approximately 1.5% at 20 aggregated runs.

These findings extend our previous observations about classification tasks to continuous variables like text length, confirming that run aggregation provides substantial benefits for consistency across different types of LLM outputs. For practitioners concerned with generating summaries of consistent length, aggregating even a small number of runs (3-5) can significantly reduce variability, with the MARD decreasing by more than half compared to single-run outputs.

In conjunction with our previous findings, these results provide strong evidence that run aggregation serves as an effective technique for enhancing LLM consistency across different output types, offering a practical approach for practitioners to significantly enhance reliability without prohibitive computational costs.

5.3 Impact of Aggregation on Classification Accuracy

The results in the previous section suggest that aggregation over multiple classification runs can improve consistency. We conduct a further analysis by exploring the impact of aggregation on accuracy (measured using F1 score) to understand how combining multiple runs affects model performance in classification tasks. For this analysis, we use the same Financial PhraseBank dataset as in Section 5.1, focusing on sentences where expert human annotators reached at least 75% agreement, using these consensus labels as ground truth, following Yoo (2024). We specifically selected cases where the 50 model runs show disagreement on labels, making aggregation sensible. At each aggregation level, we generate 50 synthetic runs and merge them with human labels. We find that the sentences in the resulting sample are highly imbalanced (with most sentences assigned neutral labels by humans), so we apply SMOTE (Synthetic Minority Over-sampling Technique) to create a balanced sample. We then calculate the weighted F1 score by class using expert labels as ground truth.

We plot the results for the two models in Panel C of Figure 2. The figure reveals striking differences in how aggregation affects the results from the two models. For GPT-3.5-turbo (left panel), F1 score surprisingly declines as more runs are aggregated, dropping from approximately 0.29 to below 0.23, falling beneath the baseline of 0.28 for no aggregation. This suggests that for this model, aggregating multiple runs actually diminishes classification accuracy. In contrast, GPT-4o-mini (right panel) shows remarkable improvement with aggregation, jumping from about 0.75 to 0.84 after just a few aggregated runs before stabilizing. The error bars, representing one standard deviation above and below the mean, are noticeably larger for GPT-3.5-turbo, indicating higher performance variability compared to the more consistent GPT-4o-mini.

These findings have important implications when considered alongside our previous results. While both models demonstrate high consistency overall (as shown in Figure 1), their responses to run aggregation differ dramatically. This suggests that although LLMs can surpass human annotators in consistency, the effectiveness of aggregation techniques varies significantly by model type. The substantial performance gain for GPT-4o-mini indicates that newer models may better leverage collective predictions, while older models like GPT-3.5-turbo may develop more rigid classification boundaries that aggregation cannot improve. This has implications when users decide whether to aggregate over runs to improve consistency, as this comes at a cost or benefit of decreased/improved accuracy, depending on model types, with newer models seeming to benefit from aggregation.

5.4 Impact on Downstream Task

Current studies in finance and accounting increasingly incorporate variables generated from LLM outputs in downstream regression analyses. Our final research question (RQ5) examines whether and how variation in LLM outputs affects the reliability of such statistical analyses. To provide preliminary evidence, we implement a simulation-based methodology that quantifies how variability in LLM-generated MD&A summaries propagates into subsequent regression analyses.

Our test evaluates how inconsistencies in summary length impact coefficient estimates, statistical significance determinations, and overall inference reliability. To do so, we first scale the length of each summary by the total length of the MD&A, generating an inverse measure of disclosure bloatedness, similar to Kim et al. (2024a). We then assess consistency using a procedure best characterized as a Monte Carlo simulation with bootstrapping and leave-one-out validation (Efron and Tibshirani 1994, Shao and Tu 2012, Hammersley 2013).

Using data from 50 runs collected in our earlier consistency analyses, we generate 1,001 synthetic runs by randomly selecting the length from the 50 runs for each document in our sample. We randomly designate one synthetic run as the "ground truth" and simulate an outcome variable with a known relationship to summary length. We then estimate regression models using the remaining 1,000 synthetic runs, comparing their statistical properties to their true values. This leave-one-out approach allows us to precisely quantify how LLM inconsistency affects key regression outputs, including coefficient bias, t-statistic distributions, and error rates in hypothesis testing. To ensure robustness, we repeat this process for 10,000 iterations, conducting a total of 10 million regressions.

Our simulation employs the following parameters to create a realistic scenario for evaluating how measurement variability affects statistical inference:

- **X effect: 0.2** - A substantial effect size for the control variable, making it relatively easy to detect. X is standardized to have a mean of zero and a standard deviation of one.
- **Length effect: 0.005** - An intentionally small effect size that makes it challenging to detect the length effect consistently. This small magnitude mirrors realistic scenarios where document length effects may be subtle but still meaningful. The scaled length is standardized to have a mean of zero and standard deviation of one.
- **Confounding: 0.5** - Indicates a moderate correlation between control variables and length, while avoiding multicollinearity issues.
- **Signal-to-Noise Ratio (SNR): 0.5**. This sets the error term to have a standard deviation that is twice the combined signal (X effect multiplied by X's standard deviation + Length effect multiplied by length's standard deviation). This setup creates a challenging but realistic environment for statistical inference.
- **Number of iterations:** 10,000.
- **Number of synthetic runs per iteration:** 1,001.
- **Standard errors:** Robust standard errors.

We choose the parameters to make it hard to detect the effects of length, which is typical in empirical research in accounting and finance, where factors such as noise, measurement errors, and other confounding factors often obscure effects. The R-squared of these regressions is approximately 0.2, which is in line with most empirical studies in accounting and finance.

As shown in Figure 4, our simulation reveals several important findings. Panel A demonstrates that despite the variability in document lengths across runs, coefficient estimates remain remarkably stable and appear unbiased. The distribution of estimated coefficients (blue) closely aligns with the distribution of true values (red), with minimal shift or distortion. The right side of Panel A shows a strong correlation between true and estimated coefficients, with points clustering tightly around the perfect agreement line.

Panel B illustrates a similar pattern for t-statistics, with the distributions of estimated and true t-statistics closely overlapping. This suggests that statistical significance determinations remain reliable despite the underlying variability in document lengths due to the randomness from LLMs. The fitted trend line in the right plot almost perfectly tracks the perfect agreement line, indicating minimal systematic bias in statistical significance testing.

Most notably, Panel C reveals that statistical inference remains highly reliable despite LLM output inconsistency. The pie chart shows that 97.35% of all inferences drawn about the significance of length are correct, with Type I and Type II errors occurring at only 1.28% and 1.38%, respectively. To assess whether variation may result in directional errors, we created a reliability heatmap. This further demonstrates that when the length effect is non-significant, correct sign identification occurs 86.15% of the time, while significant effects maintain proper sign direction in virtually all cases (0% incorrect sign when significant). The overall percentage of correct coefficient sign identification stands at 92.41%. The chart clearly indicates that when a significant result is found, it is essentially impossible for the result to have the wrong direction.[3]

[3] The slight numerical difference between the pie chart's 97.35% correct inference rate and the heatmap's combined correct cells (92.4%) stems from their different analytical approaches. The pie chart classifies results based solely on significance testing correctness (whether the test correctly identified an effect as significant or non-significant), while the heatmap introduces an additional dimension of sign correctness. In this simulation, Type I errors (1.28%) represent cases where a null effect is incorrectly found to be significant, while the 0% "Significant + Incorrect Sign" in the heatmap indicates that when significant effects are detected, they never have the wrong sign direction. These complementary metrics provide a comprehensive view of statistical reliability across different aspects of inference.

When we aggregate results across multiple runs, the inference becomes even more reliable. At an aggregation level of three runs, the percentage of correct inference increases to 98.49%, and the percentage of correct coefficient signs rises to 95.45%.

These results suggest that while LLM summarization introduces measurable inconsistency in document length, the impact on downstream statistical analyses is minimal. Regression coefficients remain unbiased, t-statistics maintain their expected distribution, and inference validity is preserved at high rates. This provides reassurance that LLM-based document processing can be reliably incorporated into quantitative research pipelines in finance and accounting, even when analyzing subtle effects where measurement precision is important.

6. Discussion and Conclusion

This study provides the first comprehensive assessment of the consistency and reproducibility of outputs from commonly used LLMs across common finance and accounting research tasks. Through extensive experimentation involving 50 independent runs for each task and generating over 3.4 million outputs, we have established several key findings regarding LLM reliability in financial text analysis.

Our results demonstrate that LLM output consistency varies substantially across different tasks. Binary classification and sentiment analysis achieve near-perfect reproducibility, while more complex tasks exhibit greater variability. For example, multi-class classification shows moderate consistency and numerical predictions display substantial variation for less advanced models like GPT-3.5-turbo. Contrary to expectations, more advanced models do not consistently demonstrate superior reproducibility. While GPT-4o achieves exceptional consistency in numerical predictions (MARD = 2.05%), GPT-4o-mini unexpectedly outperforms GPT-3.5-turbo in a multi-class classification task. For textual outputs, we observe high semantic consistency (mean cosine similarity > 0.94) alongside moderate variation in structural features such as length and tone.

Our analysis of the Financial PhraseBank dataset reveals that LLMs outperform human annotators in consistency for over 50% of sentences, with humans outperforming LLMs in fewer than 9% of cases. LLMs maintain high agreement levels (96.8-98.3%) even for sentences where human annotators significantly disagree. Additionally, aggregation strategies like majority voting for categorical outputs and averaging for continuous predictions can substantially improve consistency, with the most dramatic gains occurring within aggregation levels of 3-5 runs.

These findings have significant methodological implications. Researchers should ideally report consistency metrics alongside performance metrics when using LLMs, consider task complexity when applying these models, and conduct task-specific evaluations of both performance and consistency when selecting appropriate models. Our simulation of downstream regression analyses shows that despite output variability, statistical inference remains highly reliable (97.35% correct inferences) with minimal Type I and Type II errors (approximately 1.3% each).

Our additional analyses suggest that implementing aggregation over multiple runs through majority voting or averaging can achieve substantial consistency improvements, and that even moderate aggregation can improve the accuracy of classification tasks for newer models. Researchers analyzing textual outputs should focus primarily on semantic content rather than structural features, as the former demonstrates much higher consistency across runs. For numerical predictions, GPT-4o offers substantially superior consistency compared to less advanced models, potentially justifying its higher cost in contexts where prediction stability is crucial.

While comprehensive, our study has limitations that suggest future research directions. We focused exclusively on OpenAI models using zero-shot prompting, and future studies should extend this analysis to other LLM providers and prompting strategies. Additional finance and accounting applications beyond our five key tasks, such as stock return prediction, risk assessment, and fraud detection, also warrant investigation. Furthermore, examining how consistency patterns evolve as models are updated would provide insights into the trajectory of reliability improvements in LLM technology. We also call for more research into whether aggregation over multiple runs can increase accuracy in addition to consistency.

Our findings have important implications for addressing what we term "G-hacking"—the selective reporting of favorable outcomes from multiple runs of a Generative LLM. The high consistency we observe across most tasks suggests that G-hacking risks are relatively low for many finance and accounting applications, particularly for classification and sentiment analysis. However, tasks with higher output variability, such as numerical predictions using less advanced models, remain vulnerable to selective reporting practices. To mitigate this risk, we recommend that researchers adopt transparent reporting standards that include: (1) disclosing the number of model runs conducted, (2) specifying whether and how results were aggregated across runs, (3) reporting consistency metrics alongside performance metrics, and (4) conducting sensitivity analyses where appropriate. These practices, combined with the aggregation strategies validated in our study, provide a practical framework for maintaining research integrity in the emerging field of LLM-based research in finance and accounting.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process:

In preparing this manuscript, the author(s) used ChatGPT and Claude 3.7 to enhance language and readability. Following the use of these tools, the author(s) carefully reviewed and edited the content to ensure accuracy and take full responsibility for the final version of the manuscript.

References

Aguda T, Siddagangappa S, Kochkina E, Kaur S, Wang D, Smiley C, Shah S (2024) Large language models as financial data annotators: A study on effectiveness and efficiency. (March 26) http://arxiv.org/abs/2403.18152.

Ahmed H, Lofstead J (2022) Managing randomness to enable reproducible machine learning. *Proceedings of the 5th International Workshop on Practical Reproducible Evaluation of Computer Systems* P-RECS '22. (Association for Computing Machinery, New York, NY, USA), 15–20.

Alonso-Robisco A, Carbó JM (2023) Analysis of CBDC narrative by central banks using large language models. *Finance Research Letters* 58:104643.

Álvarez-Díez S, Baixauli-Soler JS, Kondratenko A, Lozano-Reina G (2024) Dividend announcement and the value of sentiment analysis. *Journal of Management Analytics* 0(0):1–21.

Artstein R, Poesio M (2008) Inter-coder agreement for computational linguistics. *Computational Linguistics* 34(4):555–596.

Bae J, Berger AN, Choi HS, Kim HH (2024) Bank sentiment and loan loss provisioning. (March 2) https://papers.ssrn.com/abstract=4745996.

Bai J (Jianqiu), Boyson NM, Cao Y, Liu M, Wan C (2023) Executives vs. Chatbots: Unmasking insights through human-AI differences in earnings conference Q&A. (June 15) https://papers.ssrn.com/abstract=4480056.

Bernard D, Blankespoor E, de Kok T, Toynbee S (2023) Confused readers: A modular measure of business complexity. (June 15) https://papers.ssrn.com/abstract=4480309.

Blankespoor E, Croom J, Grant SM (2024) Generative AI and investor processing of financial information. (December 12) https://papers.ssrn.com/abstract=5053905.

Bochkay K, Brown SV, Leone AJ, Tucker JW (2022) Textual analysis in accounting: What's next? *Contemporary Accounting Research* (Forthcoming).

Bond SA, Klok H, Zhu M (2025) Large language models and financial market sentiment. (January 26) https://papers.ssrn.com/abstract=4584928.

Bozanic Z, Roulstone DT, Van Buskirk A (2018) Management earnings forecasts and other forward-looking statements. *Journal of Accounting and Economics* 65(1):1–20.

Breitung C, Kruthof G, Müller S (2023) Contextualized sentiment analysis using large language models. (October 27) https://papers.ssrn.com/abstract=4615038.

Brown SV, Hinson LA, Tucker JW (2024) Financial statement adequacy and firms' MD&A disclosures. *Contemporary Accounting Research* 41(1):126–162.

Chen AY, Zimmermann T (2022) Open source cross-sectional asset pricing. *CFR* 11(2):207–264.

Chen B, Wu Z, Zhao R (2023) From fiction to fact: the growing role of generative AI in business and finance. *Journal of Chinese Economic and Business Studies* 21(4):471–496.

Chen J, Tang G, Zhou G, Zhu W (2023) ChatGPT, stock market predictability and links to the macroeconomy. (July 31) https://papers.ssrn.com/abstract=4660148.

Choi GY, Kim A (2023) Economic footprints of tax audits: A generative AI-driven approach. (December 15) https://papers.ssrn.com/abstract=4645865.

Comlekci İ, Unal S, Ozer A, Oncu MA (2023) Can AI technologies estimate financials accurately? A research on Borsa Istanbul with ChatGPT. (April 8) https://papers.ssrn.com/abstract=4545954.

Dasgupta S, Li EXN, Wu S (2023) Inferring financial flexibility: Do actions speak louder than words? (June 17) https://papers.ssrn.com/abstract=4482620.

Dong MM, Stratopoulos TC, Wang VX (2024) A scoping review of ChatGPT research in accounting and finance. *International Journal of Accounting Information Systems* 55:100715.

Efron B, Tibshirani RJ (1994) *An introduction to the bootstrap* (Chapman and Hall/CRC, New York).

Fišar M, Greiner B, Huber C, Katok E, Ozkes AI (2024) Reproducibility in Management Science. *Management Science* 70(3):1343–1356.

Fleiss JL (1971) Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5):378–382.

Gilardi F, Alizadeh M, Kubli M (2023) ChatGPT outperforms crowd-workers for text-annotation tasks. *Proc. Natl. Acad. Sci. U.S.A.* 120(30):e2305016120.

Glasserman P, Lin C (2023) Assessing look-ahead bias in stock return predictions generated by GPT sentiment analysis. (September 28) https://papers.ssrn.com/abstract=4586726.

Gow ID (2025) The Elephant in the Room: p-hacking and Accounting Research. *Accounting, Economics, and Law: A Convivium* 15(1):81–98.

Gundersen OE, Kjensmo S (2018) State of the art: Reproducibility in artificial intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence* 32(1).

Hail L, Lang M, Leuz C (2020) Reproducibility in accounting research: Views of the research community. *Journal of Accounting Research* 58(2):519–543.

Hammersley J (2013) *Monte Carlo methods* (Springer Science & Business Media).

Hansen AL, Kazinnik S (2024) Can ChatGPT decipher fedspeak? (April 10) https://papers.ssrn.com/abstract=4399406.

Harvey CR (2019) Replication in financial economics. (July 29) https://papers.ssrn.com/abstract=3409466.

Hou K, Xue C, Zhang L (2020) Replicating anomalies. *The Review of Financial Studies* 33(5):2019–2133.

Hu N, Liang P, Yang X (2023) Whetting all your appetites for financial tasks with one meal from GPT? A comparison of GPT, FinBERT, and dictionaries in evaluating sentiment analysis. (July 26) https://papers.ssrn.com/abstract=4426455.

Huang AH, Wang H, Yang Y (2023) FinBERT: A large language model for extracting information from financial text. *Contemporary Accounting Research* 40(2):806–841.

Jensen TI, Kelly B, Pedersen LH (2023) Is there a replication crisis in finance? *The Journal of Finance* 78(5):2465–2518.

Jia N, Li N, Ma G, Xu D (2024) Corporate responses to generative AI: Early evidence from conference calls. (February 23) https://papers.ssrn.com/abstract=4736295.

Kim A, Muhn M, Nikolaev VV (2024a) Bloated disclosures: Can ChatGPT help investors process information? (August 27) https://papers.ssrn.com/abstract=4425527.

Kim A, Muhn M, Nikolaev VV (2024b) From transcripts to insights: Uncovering corporate risks using generative AI. (October 26) https://papers.ssrn.com/abstract=4593660.

Kim AG, Muhn M, Nikolaev VV (2024) Financial statement analysis with large language models. (November 7) https://papers.ssrn.com/abstract=4835311.

Kirtac K, Germano G (2024) Sentiment trading with large language models. (March 21) https://papers.ssrn.com/abstract=4706629.

de Kok T (2025) ChatGPT for textual analysis? How to use generative LLMs in accounting research. *Management Science*.

Krippendorff K (2004) Reliability in content analysis: Some common misconceptions and recommendations. *Human Communication Research* 30(3):411–433.

Krippendorff K (2011) Computing Krippendorff's alpha-reliability.

Kuroki Y, Manabe T, Nakagawa K (2023) Fact or opinion? – Essential value for financial results briefing. (April 27) https://papers.ssrn.com/abstract=4430511.

Leippold M (2023) Sentiment spin: Attacking financial sentiment with GPT-3. *Finance Research Letters* 55:103957.

Levy B (2024) Caution ahead: Numerical reasoning and look-ahead bias in AI models. (December 25) https://papers.ssrn.com/abstract=5082861.

Li EX, Tu Z, Zhou D (2024) The promise and peril of generative AI: Evidence from GPT-4 as sell-side analysts. (December 1) https://papers.ssrn.com/abstract=4480947.

Li F (2008) Annual report readability, current earnings, and earnings persistence. *Journal of Accounting and Economics* 45(2):221–247.

Li F (2010) The information content of forward-looking statements in corporate filings—A naïve Bayesian machine learning approach. *Journal of Accounting Research* 48(5):1049–1102.

Li T, Peng Q, Yu L (2023) ESG considerations in acquisitions and divestitures: Corporate responses to mandatory ESG disclosure. (May 26) https://papers.ssrn.com/abstract=4376676.

Linnainmaa JT, Roberts MR (2018) The history of the cross-section of stock returns. *The Review of Financial Studies* 31(7):2606–2649.

Lopez-Lira A, Tang Y (2024) Can ChatGPT forecast stock price movements? Return predictability and large language models. (April 14) https://papers.ssrn.com/abstract=4412788.

Loughran T, McDonald B (2011) When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance* 66(1):35–65.

Loughran T, McDonald B (2014) Measuring readability in financial disclosures. *The Journal of Finance* 69(4):1643–1671.

Malo P, Sinha A, Takala P, Korhonen P, Wallenius J (2013) Good debt or bad debt: Detecting semantic orientations in economic texts. (July 23) http://arxiv.org/abs/1307.5336.

McHugh ML (2012) Interrater reliability: The Kappa statistic. *Biochem Med (Zagreb)* 22(3):276–282.

Menkveld AJ, Dreber A, Holzmeister F, Huber J, Johannesson M, Kirchler M, NEUSÜß S, et al. (2024) Nonstandard errors. *The Journal of Finance* 79(3):2339–2390.

Muslu V, Radhakrishnan S, Subramanyam KR, Lim D (2014) Forward-looking MD&A disclosures and the information environment. *Management Science* 61(5):931–948.

Pérignon C, Akmansoy O, Hurlin C, Dreber A, Holzmeister F, Huber J, Johannesson M, et al. (2024) Computational reproducibility in finance: Evidence from 1,000 tests. *The Review of Financial Studies* 37(11):3558–3593.

Pineau J, Vincent-Lamarre P, Sinha K, Lariviere V, Beygelzimer A, d'Alche-Buc F, Fox E, Larochelle H (2021) Improving reproducibility in machine learning research. *Journal of Machine Learning Research* 22(164):1–20.

Raff E (2019) A step toward quantifying independently reproducible machine learning research. (September 14) http://arxiv.org/abs/1909.06674.

Reiss MV (2023) Testing the reliability of ChatGPT for text annotation and classification: A cautionary remark. (April 17) http://arxiv.org/abs/2304.11085.

Rossi L, Harrison K, Shklovski I (2024) The problems of LLM-generated data in social science research. *Sociologica* 18(2):145–168.

Sarkar SK, Vafa K (2024) Lookahead bias in pretrained language models. (June 28) https://papers.ssrn.com/abstract=4754678.

Shaffer M, Wang CCY (2024) Scaling core earnings measurement with large language models. (October 8) https://papers.ssrn.com/abstract=4979501.

Shao J, Tu D (2012) *The Jackknife and bootstrap* (Springer Science & Business Media).

Smales LA (2023) Classification of RBA monetary policy announcements using ChatGPT. *Finance Research Letters* 58:104514.

Sturua S, Mohr I, Akram MK, Günther M, Wang B, Krimmel M, Wang F, et al. (2024) jina-embeddings-v3: Multilingual embeddings with task LoRA. (September 19) http://arxiv.org/abs/2409.10173.

Törnberg P (2023) ChatGPT-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. (April 13) http://arxiv.org/abs/2304.06588.

Wang J, Wang VX (2024) Leveraging large language models to democratize access to costly financial datasets for academic research. (November 1) https://papers.ssrn.com/abstract=5012660.

Wang M (2023) Heads I win, tails it's chance: Mutual fund performance self-attribution. (July 29) https://papers.ssrn.com/abstract=4553526.

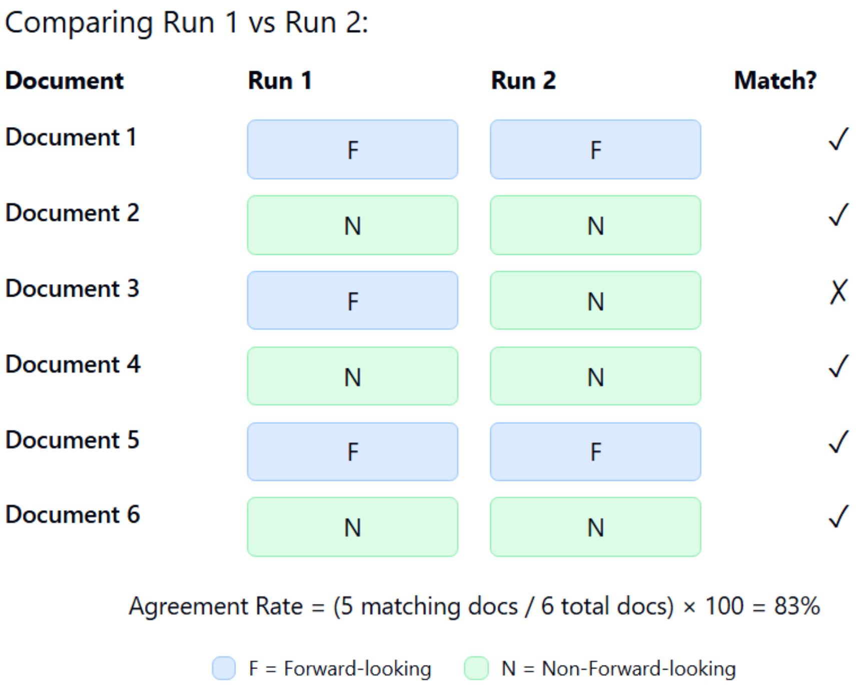
Wu Z, Dong Y, Li Y, Shi B (2023) Unleashing the power of text for credit default prediction: Comparing human-generated and AI-generated texts. (June 6) https://papers.ssrn.com/abstract=4601317.

Yoo M (2024) How much should we trust large language model-based measures for accounting and finance research? (October 10) https://papers.ssrn.com/abstract=4983334.

Appendix A
Agreement Metric Calculation Illustration

This appendix illustrates the calculation of two key agreement metrics used in our analysis: run-pair agreement and document-wise agreement. Both metrics are derived from pairwise comparisons but differ in their unit of analysis.

Panel A: Run-pair agreement



Comparing Run 1 vs Run 2:

Document	Run 1	Run 2	Match?
Document 1	F	F	✓
Document 2	N	N	✓
Document 3	F	N	✗
Document 4	N	N	✓
Document 5	F	F	✓
Document 6	N	N	✓

Agreement Rate = (5 matching docs / 6 total docs) × 100 = 83%

🟦 F = Forward-looking 🟩 N = Non-Forward-looking

Run-pair agreement assesses classification consistency between pairs of runs across all documents. The above figure shows this calculation by comparing Run 1 and Run 2 across six documents:

- Documents 1 and 5: Both runs classify as Forward-looking (F).
- Documents 2, 4, and 6: Both runs classify as Non-Forward-looking (N).
- Document 3: Run 1 classifies as F, whereas Run 2 as N (disagreement).

Using the formula shown below:

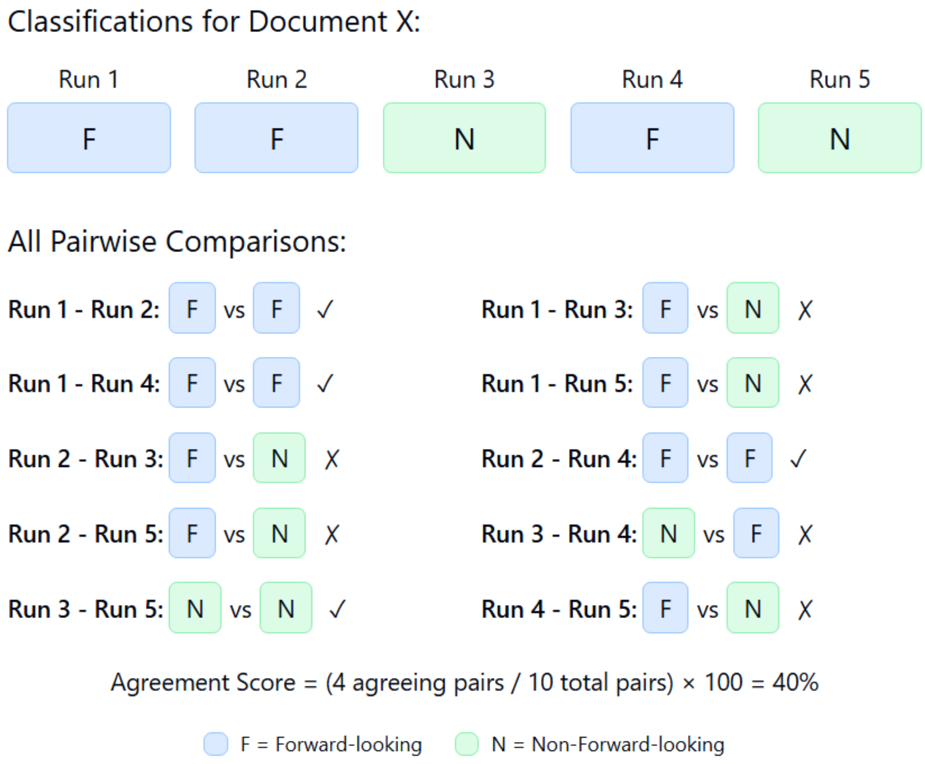
$$\text{Run-pair Agreement} = \left(\frac{\text{Number of matching documents}}{\text{Total documents compared}}\right) \times 100$$

- Matching classifications = 5 documents
- Total documents = 6
- Agreement rate = (5/6) × 100 = 83%

For the complete analysis, this calculation is performed for all 1,225 possible run pairs. The mean run-pair agreement represents the average agreement rate across all these pairs, providing a measure of overall classification stability.

Appendix A
Agreement Metric Calculation Illustration (Continued)

Panel B: Document-wise agreement



Classifications for Document X:

Run 1	Run 2	Run 3	Run 4	Run 5
F	F	N	F	N

All Pairwise Comparisons:

Run 1 - Run 2: F vs F ✓

Run 1 - Run 3: F vs N ✗

Run 1 - Run 4: F vs F ✓

Run 1 - Run 5: F vs N ✗

Run 2 - Run 3: F vs N ✗

Run 2 - Run 4: F vs F ✓

Run 2 - Run 5: F vs N ✗

Run 3 - Run 4: N vs F ✗

Run 3 - Run 5: N vs N ✓

Run 4 - Run 5: F vs N ✗

Agreement Score = (4 agreeing pairs / 10 total pairs) × 100 = 40%

🟦 F = Forward-looking 🟩 N = Non-Forward-looking

Document-wise agreement measures classification consistency for individual documents across multiple runs. For each document, we compare classifications between all possible pairs of runs. As shown in the above figure, Document X receives classifications from five runs:

- Runs 1, 2, and 4 classify it as Forward-looking (F)
- Runs 3 and 5 classify it as Non-Forward-looking (N)

Agreement is assessed through pairwise comparisons of runs. The document-wise agreement score is calculated using the formula shown below:

$$\text{Document-wise Agreement} = \left(\frac{\text{Number of agreeing pairs}}{\text{Total number of pairs}}\right) \times 100$$

- Total number of run pairs = 10 (from 5 runs)
- Number of agreeing pairs = 4
- Agreement score = (4/10) × 100 = 40%

In our full analysis with 50 runs, each document comes with 1,225 run pairs (50 × 49 / 2), and its agreement score is calculated across all these pairs. The mean document-wise agreement reported is the average of these scores across all documents.

Appendix B
Metric Definitions

Panel A: Metric Definitions for sample size metrics

Number of documents: The total number of documents used for a task. A document can be a sentence, paragraph, longer text (e.g., a full MD&A section), or a combination of multiple data types as in the case of Q&A answering and future earnings prediction. In our terminology, a document represents a single unit of the input provided to an LLM, which generates an output based on this input in addition to the prompt.

Number of runs: The number of independent attempts performed by an LLM on each document using different random seeds (50 in our experiments).

Number of run pairs: The total number of unique pairs that can be formed from the runs, calculated as (n × (n-1))/2, where "n" is the number of runs. With 50 runs, this yields 1,225 pairs. Each pair represents a unique comparison between two different runs' outputs.

Panel B: Metric definitions for categorical variables

Class-level metrics:

Class distribution: The proportion of documents assigned to each class based on majority voting across runs. **This information is provided to shed** light **on class balance or imbalance. For example, in the case of our binary classification tas**k, a percentage of 20% for forward-looking means 20% of documents are classified as forward-looking in the majority of runs.

Class-specific agreements: For each class, the average agreement rate across documents that received that classification in any run. For example, the rate may be 98% for **forward-looking** and 95% for **non-forward looking.**

Inter-rater and run-level agreements:

Fleiss' Kappa: A statistical measure assessing agreement between multiple runs while accounting for chance agreement. Ranges from -1 to 1, where 1 indicates perfect agreement and 0 indicates chance-level agreement.

Krippendorff's Alpha: A robust agreement measure for multiple runs that accounts for missing data and measurement level. Like Fleiss's Kappa, it ranges from -1 to 1.

Run-pair Cohen's Kappa score: A pairwise Cohen's Kappa score between two runs, measuring overall consistency between them. This score ranges from -1 to 1, where higher values indicate greater consistency between runs.

Mean run-pair Cohen's Kappa score: The average of all 1,250 pairwise Cohen's Kappa scores calculated between all possible pairs of runs. This provides an overall measure of consistency across multiple runs.

Run-pair agreement: The pairwise classification agreement rate between two runs, calculated as described in Panel A of Appendix A. This measures the proportion of documents that receive identical classifications when comparing two specific runs.

Mean run-pair agreement: The average of all 1,250 pairwise classification agreement rates calculated between all possible pairs of 50 runs. This metric quantifies, on average, what percentage of documents will be classified identically when comparing any two randomly selected runs across all classes.

Document-level Agreements:

Percentage of perfect agreement (%): Percentage of documents where all 50 runs assign the same class.

Document-wise agreement (%): The percentage of run pairs (out of all 1,225 possible pairs among the 50 runs) that assign the same classification to a specific document. This measures the consistency of classification for individual documents across multiple runs. See Panel B of Appendix B for an illustration.

Mean document-wise agreement (%): The average of all document-wise run agreement percentages across all documents in a sample. This aggregate metric quantifies how often, on average, any two randomly **selected runs will agree on the classification of a document, providing an** overall **measure of classification consistency across the entire dataset.**

Majority class strength (%): The percentage of runs that assign the most frequent classification to a document. For each document, we identify which classification appears most frequently across the 50 runs, then calculate what percentage of those runs assigned that majority classification.

Mean majority class strength (%): The average of majority class strength percentages across all documents in a sample. This metric quantifies the overall dominance of majority classifications across the sample.

Classification uncertainty: The entropy of the classification distribution across runs for a document where there is disagreement in labeling. This is calculated using the Shannon entropy formula: $-\sum(p(i) \times \log_2(p(i)))$, where p(i) is the proportion of runs assigning class "i" to the document. Higher values indicate greater variability in how documents are classified across different runs, with maximum entropy occurring when classifications are evenly distributed across all possible classes. The entropy ranges from 0 (when all runs assign the same classification) to $\log_2(n)$ (when classifications are evenly distributed across all "n" possible classes).

Mean classification uncertainty: The average of classification uncertainty (entropy) values across all documents which have any disagreement over the 50 runs. This metric provides an overall measure of classification variability in a sample, excluding documents with perfect agreement.

Panel C: Metric Definitions for Continuous Variables

Run-Level Metrics:

ICC2: Intraclass Correlation Coefficient (type 2), which measures the degree of consistency between ratings from different runs across all documents. The ICC(2) value indicates what proportion of the total variance in document ratings is attributable to true differences between the documents themselves, rather than differences between runs or random error. Values range from 0 to 1, with higher values indicating greater consistency.

Concordance correlation: The pairwise Concordance Correlation Coefficient (CCC) between two runs, measuring both precision and accuracy in agreement. Values range from -1 to 1, with 1 indicating perfect agreement, 0 indicating no correlation, and -1 indicating perfect negative agreement.

Mean concordance correlation: The average of all 1,225 pairwise Concordance Correlation Coefficients (CCC) calculated between all possible pairs among the 50 runs.

Spearman correlation: The Spearman rank correlation coefficient between two runs, measuring the strength and direction of monotonic relationship between classifications or scores. Values range from -1 to 1, with higher values indicating stronger rank-order consistency.

Median Spearman correlation: The average of all 1,225 pairwise Spearman correlation coefficients calculated between all possible pairs among the 50 runs.

Run-pair MARD (%): The Mean Absolute Relative Differences (MARD) between two runs, expressed as a percentage. For each document, the relative difference is calculated as: $|A-B|/((|A|+|B|)/2)$ *100, where A and B are the values from the two runs and $|x|$ indicates absolute value. These differences are then averaged across all documents in a sample. Lower values indicate smaller variations between runs.

Mean run-pair MARD (%): The average run-pair MARDs between all 1,225 run-pair MARD values calculated between all possible pairs among the 50 runs.

Document-Level Metrics

Documents with identical output (%): The percentage of the documents in a sample that have the same output from the model across the 50 runs, e.g., same length for summarization tasks and same amount for point estimates of future earnings.

Document-wise MARD (%): The Mean Absolute Relative Difference for a document calculated across all possible run pairs. The calculation uses the formula |A-B|/((|A|+|B|)/2)*100 for each run pair, where A and B are the values produced by different runs. These pair-wise differences are then averaged to give the document's MARD percentage.

Mean document-wise MARD (%): The average document-wise MARDs across all documents in a sample.

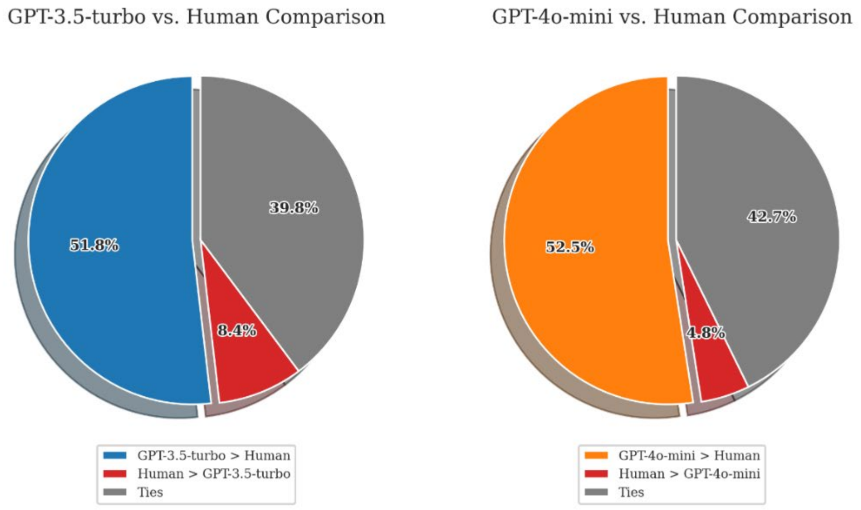
Panel D: Metric definition for text outputs

Run-pair similarity: Average cosine similarity between two runs, calculated using text embeddings generated by the *jina-embeddings-v3* model for all corresponding documents across the runs.

Document-level similarity: Average cosine similarity for a document across all possible run pairs (from the 50 runs), calculated using text embeddings generated by the jina-*embeddings-v3* model.

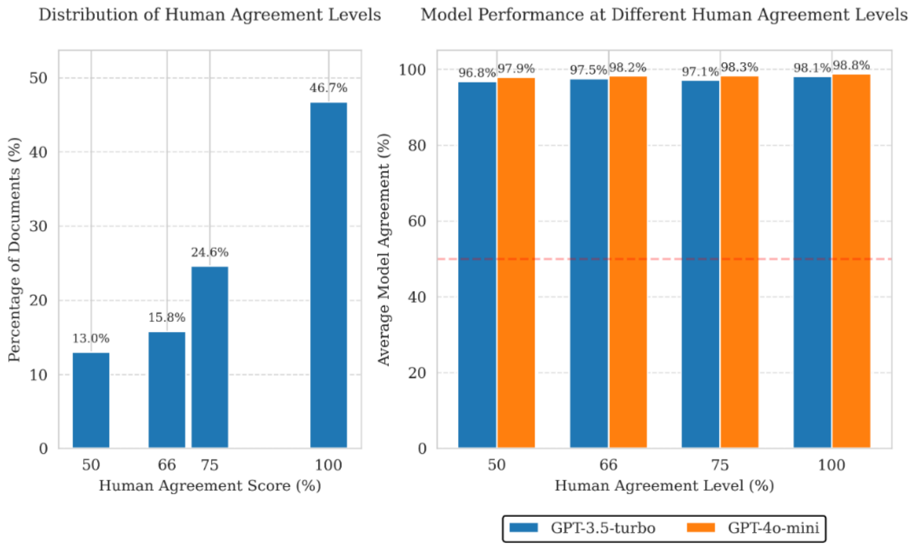
Figure 1 Comparison of LLM and Human Annotator Agreement Patterns in Sentiment Classification

Panel A: Consistency Comparison Between LLMs and Human Annotators



Note: The left pie chart compares agreement levels between GPT-3.5-turbo and human annotators. The blue slice indicates that GPT-3.5-turbo outperforms human annotators in 51.8% of sentences. The right pie chart provides a similar comparison between GPT-4o-mini and human annotators.

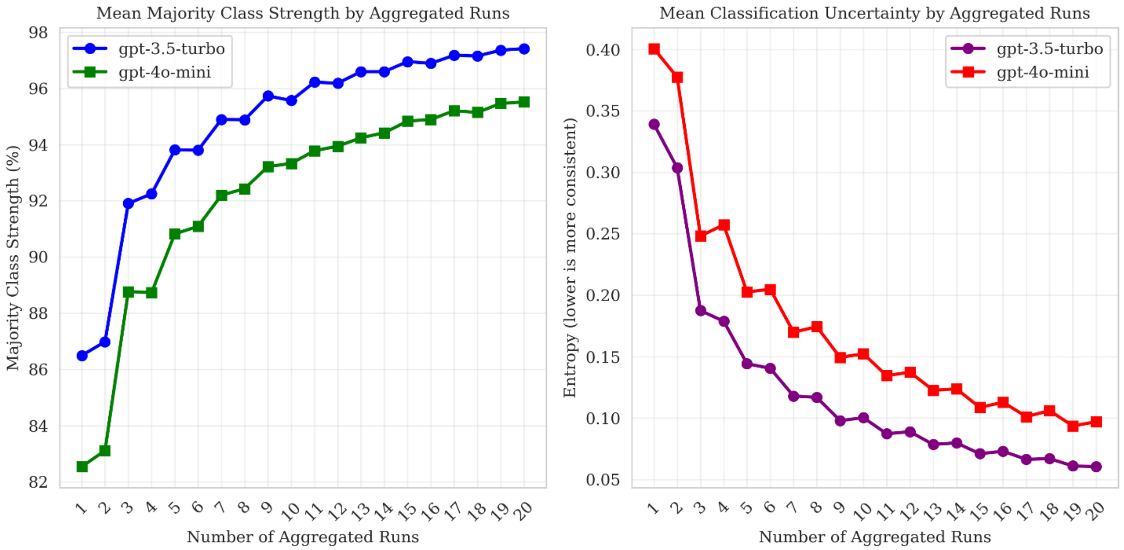
Panel B: LLM Performance Across Varying Human Agreement Levels



Note: The left bar chart displays the distribution of agreement levels among human annotators, with 46.7% of sentences achieving perfect agreement (100%), meaning all annotators assigned identical labels. The right bar chart illustrates the performance of GPT-3.5-turbo and GPT-4o-mini across different human agreement levels, showing model agreement percentages for each category.

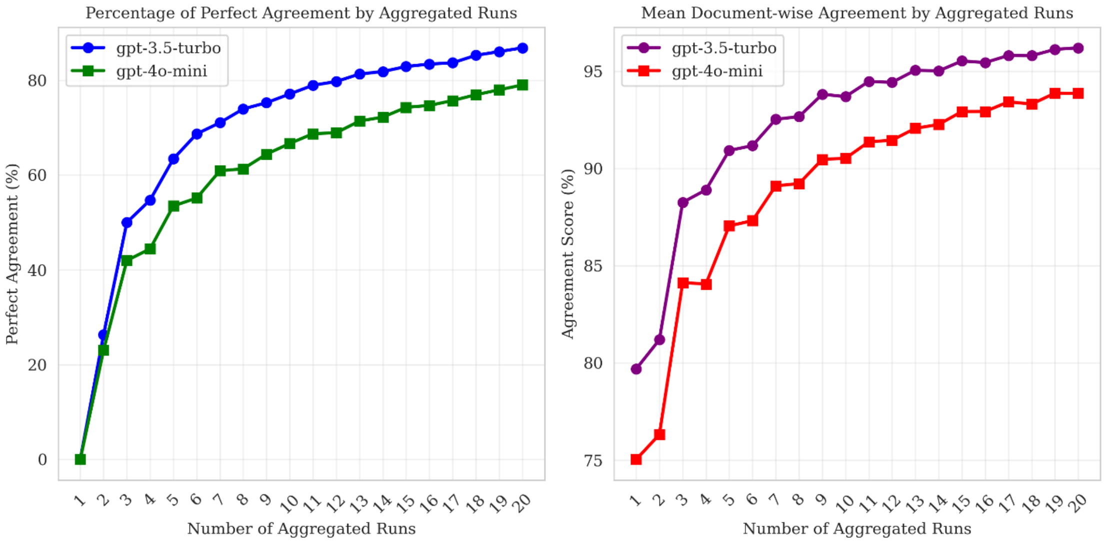
Figure 2 Effects of Run Aggregation on Classification Results

Panel A: Impact of Aggregation on Classification Strength and Uncertainty



Note: The left line chart shows the majority class strength across different aggregation levels for two GPT models, demonstrating increasing confidence as more runs are aggregated. A value of '1' indicates no aggregation, while '2' signifies that two runs are aggregated using majority voting. The right line chart displays the average classification uncertainty (entropy) across all documents at various aggregation levels for both GPT models, revealing decreasing uncertainty as aggregation increases.

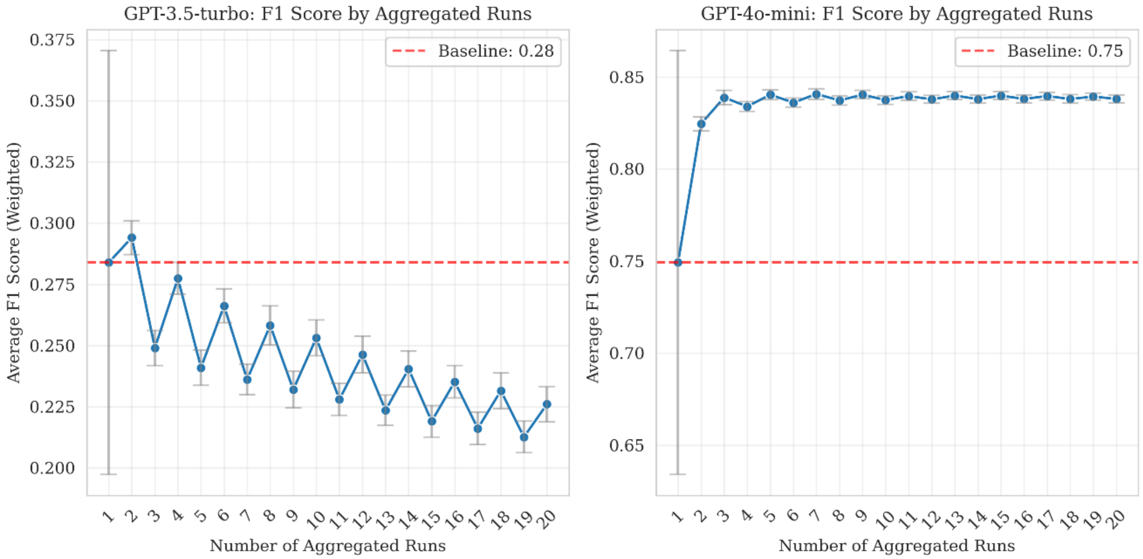
Panel B: Impact of Run Aggregation on Document-level Agreements



Note: The left line chart shows the percentage of perfect agreement across different aggregation levels for two GPT models, demonstrating increasing consistency as more runs are aggregated. A value of '1' indicates no aggregation, while '2' signifies that two runs are aggregated using majority voting. The right line chart plots the mean document-wise agreement across different aggregation levels for the two GPT models, similarly revealing improved agreement as the number of aggregated runs increases.

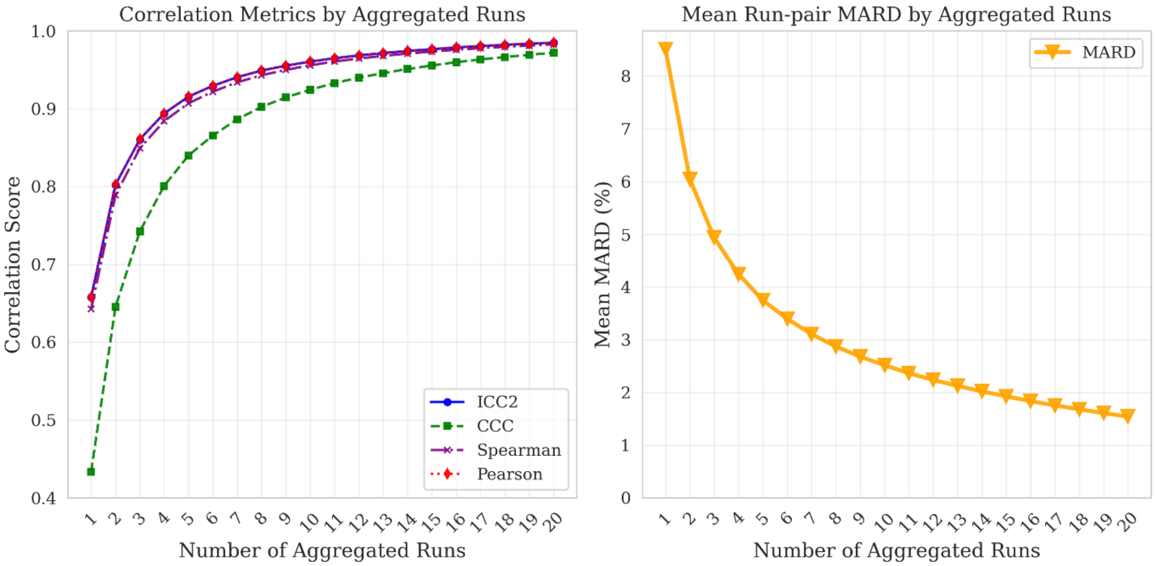
Figure 2 Effects of Run Aggregation on Classification Results (Continued)

Panel C: Impact of Run Aggregation on Accuracy (F1 Score)



Note: This panel shows weighted F1 scores by aggregation level for GPT-3.5-turbo (left) and GPT-4o-mini (right). F1 score is the harmonic mean of precision and recall (F1 = 2 × (precision × recall) / (precision + recall)), providing a balanced measure of model accuracy that accounts for both false positives and false negatives. The weighted F1 score accounts for class imbalance by calculating F1 for each class separately and averaging them weighted by class frequency. Higher F1 scores (maximum 1.0) indicate better accuracy. Error bars represent one standard deviation above and below the mean across 50 synthetic runs. Baseline values (dashed red lines) represent performance without aggregation.

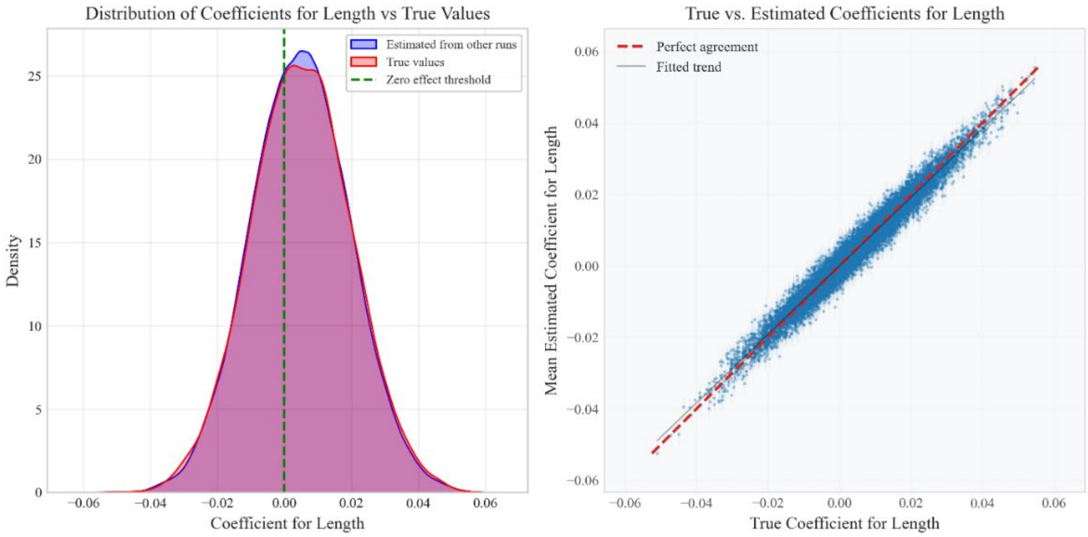
Figure 3 Effects of Run Aggregation on Length of MD&A Summaries



Note: The left line chart shows how various correlation metrics (ICC2, CCC, Spearman, and Pearson) improve as the number of aggregated runs increases from 1 to 20. The right line chart illustrates the declining Mean Run-pair MARD (%) as more runs are aggregated, indicating that variability between run pairs decreases substantially with increased aggregation.

Figure 4 Impact on Downstream Tests

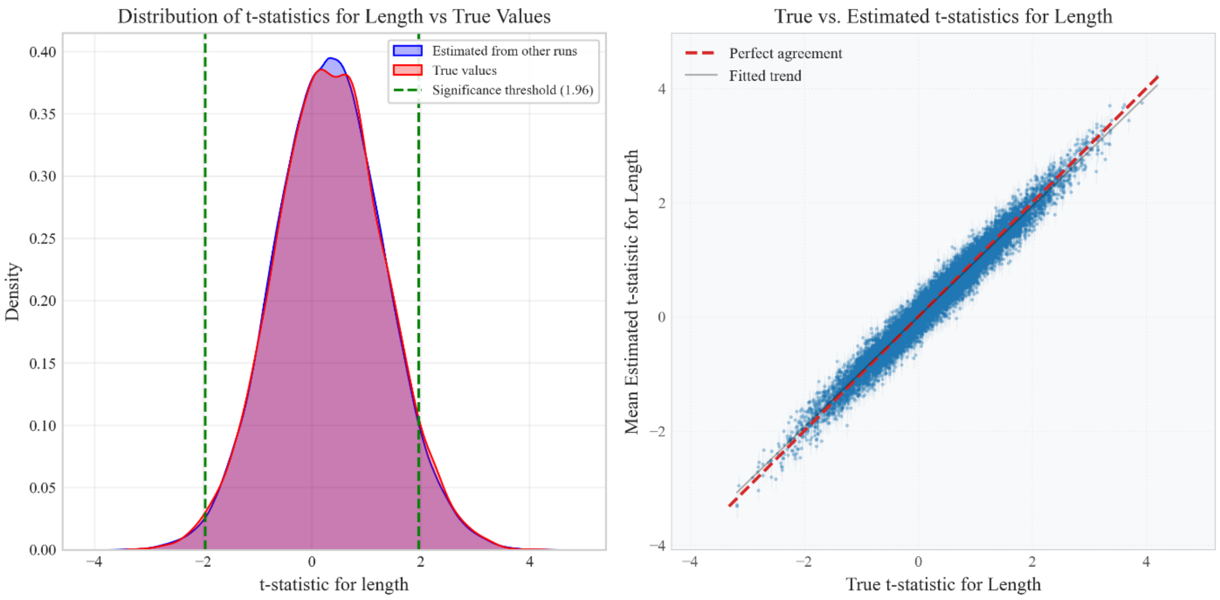
Panel A: Impacts on Coefficients



Note: The left chart displays the distribution of estimated coefficients for "Length" compared to the true values. The right chart plots true coefficients against their estimated values, with the red dashed line representing perfect agreement and the gray line showing the fitted trend. The close alignment of points along the diagonal demonstrates high agreement between true values and estimated values.

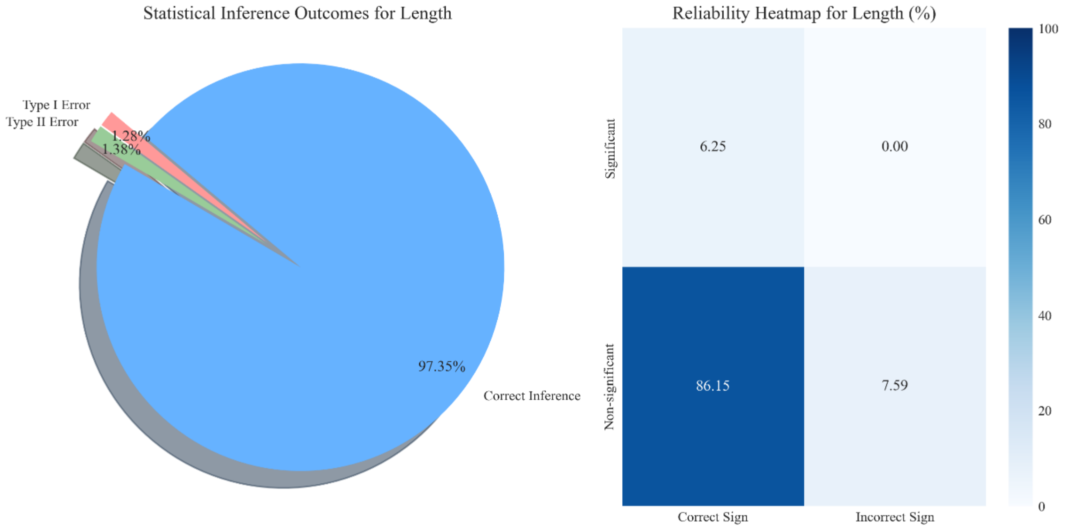
Figure 4 Impact on Downstream Tests (Continued)

Panel B: Impacts on t-statistics



Note: The left chart displays the distribution of estimated t-statistics for "Length" compared to their true values. The right chart plots true t-statistics against their estimated values, with the red dashed line representing perfect agreement and the gray line showing the fitted trend. The close alignment of points along the diagonal demonstrates high agreement between true values and estimated values.

Panel C: Impacts on Inferences



Note: This panel presents two complementary visualizations of statistical inference reliability. The left pie chart displays overall inference accuracy (97.35% correct) with Type I and Type II error rates. The right heatmap provides a more granular analysis by cross-tabulating significance status with sign correctness.

Table 1 Task Design

Panel A: Task Choice

Task	Description	Output
Classification (binary)	• Classifying forward-looking sentences from MD&As	Two labels: “Forward-looking” and “Non-forward-looking”
Classification (multi-class)	• Classifying sentences from FOMC statements	Five labels: "Hawkish", "Mostly Hawkish", "Neutral", "Mostly Dovish", or "Dovish"
Sentiment analysis	• Labelling sentences from news articles • Labelling sentences from MD&As • Labelling paragraphs from earnings call presentations • Labelling paragraphs from earnings call Q&As	Three labels: “Positive”, “Negative”, and “Neutral”
Summarization	• Summarizing MD&As • Summarizing earnings call presentations	Summaries
Text generation	• Answering questions from earnings call Q&As	Answers
Prediction	• Predicting future earnings based on comparative financial statements and MD&A summaries	• Directional prediction: “Increase” or “Decrease” • Point estimates of earnings before extraordinary items

Panel B: Source Data and Sample Size

Task	Source Data	Sample size	Unit
Classification	MD&As from 10-Ks	10,000	Sentences
Classification	FOMC statements	1,096	Sentences
Sentiment analysis	News articles	4,838	Sentences
Sentiment analysis	MD&As from 10-Ks	10,000	Sentences
Sentiment analysis	Presentation sections of earnings calls	2,000	Paragraphs
Sentiment analysis	Q&A sections of earnings calls	2,000	Paragraphs
Summarization	MD&As from 10-Ks	1,000	MD&As
Summarization	Presentation sections of earnings calls	1,000	Presentations
Text generation	Presentation summary and Q&A section	2,000	Questions
Prediction	MD&A summary and comparative statements	1,000	Firm-years

Table 2 Model Choice

Panel A: Model Characteristics

Model Name	Context Window	Max Output Tokens	Knowledge Cut-off	Input Token Cost	Output Token Cost
GPT-3.5-turbo (0125)	16,385 tokens	4,096 tokens	Sep-2021	$0.50 / million tokens	$1.50 / million tokens
GPT-4o-mini (2024-07-18)	128,000 tokens	16,384 tokens	Oct-2023	$0.15 / million tokens	$0.60 / million tokens
GPT-4o (2024-11-20)	128,000 tokens	16,384 tokens	Oct-2023	$2.50 / million tokens	$10.00 / million tokens

Source: https://platform.openai.com/docs/models/gp and https://openai.com/api/pricing/

Panel B: Models for Specific Tasks

Task	Models
Classification (binary)	GPT-3.5-turbo-0125 GPT-4o-mini-2024-07-18
Classification (multi-class)	GPT-3.5-turbo-0125 GPT-4o-mini-2024-07-18
Sentiment analysis	GPT-3.5-turbo-0125 GPT-4o-mini-2024-07-18
Summarization	GPT-4o-mini-2024-07-18
Text generation	GPT-3.5-turbo-0125 GPT-4o-mini-2024-07-18
Prediction	GPT-3.5-turbo-0125 GPT-4o-mini-2024-07-18 GPT-4o-2024-11-20

Table 3 Evaluation Metrics

Panel A: Metrics for Categorical Outputs

Metric Category	Metric Name
Overall inter-rater agreements	Fleiss' Kappa
	Krippendorff's Alpha
Run-level agreements	Run-pair Cohen's Kappa score
	Run-pair agreement (%)
Document-level agreements	Percentage of perfect agreement (%)
	Document-wise agreement (%)
	Majority class strength (%)
	Classification uncertainty (%)

Panel B: Metrics for Continuous Outputs

Metric Category	Metric Name
Overall inter-run reliability	Intraclass Correlation Coefficient (ICC2)
Correlation metrics	Concordance correlation
	Spearman correlation
Run-level variation	Run-pair MARD (%)
Document-level variation	Documents with identical output (%)
	Document-wise MARD (%)

Panel C: Evaluation Metrics for Text Outputs

Text Characteristic	Description
Length	Word count (evaluated using continuous metrics including ICC2, CCC, Spearman correlation, and MARD)
Tone	Tone measured using Loughran-McDonald finance dictionary and FinBERT
Semantic similarity	Cosine similarity calculated from text embeddings generated by using a state-of-the-art embedding model, namely, *jina-embeddings-v3*

Table 4 Summary Consistency Results for Classification Tasks

	Binary (FLS)		Multi-class (FOMC)	
Metric	**GPT-4o-mini**	**GPT-3.5-turbo**	**GPT-4o-mini**	**GPT-3.5-turbo**
Overall Inter-rater Agreements				
Fleiss' Kappa	0.97	0.93	0.86	0.91
Krippendorff's Alpha	0.97	0.93	0.86	0.91
Run-level Agreements				
Mean run-pair Cohen's Kappa	0.97	0.93	0.86	0.91
Mean run-pair agreement (%)	99.00	98.07	88.93	92.96
Document-level Agreements				
Percentage of perfect agreement (%)	96.18	92.17	55.66	65.33
Mean document-wise agreement (%)	99.00	98.07	88.93	92.96
Mean majority class strength (%)	99.30	98.67	92.26	95.32
Mean classification uncertainty	0.40	0.38	0.40	0.34

Note: See Appendix B for metric definitions.

Table 5 Summary Consistency Results for Sentiment Analysis Tasks

	News		MD&A		Presentation		Q&A	
Metric	**4oM**	**3.5T**	**4oM**	**3.5T**	**4oM**	**3.5T**	**4oM**	**3.5T**
Overall Inter-rater Agreements								
Fleiss' Kappa	0.96	0.94	0.97	0.95	0.98	0.96	0.97	0.95
Krippendorff's Alpha	0.96	0.94	0.97	0.95	0.98	0.96	0.97	0.95
Run-level Agreements								
Mean run-pair Cohen's Kappa	0.96	0.94	0.97	0.95	0.98	0.96	0.97	0.95
Mean run-pair agreement (%)	97.75	96.38	97.97	96.92	98.71	97.47	98.55	97.49
Document-level Agreements								
Percentage of perfect agreement (%)	88.28	81.09	92.46	88.53	95.05	90.65	93.80	90.10
Mean document-wise agreement (%)	97.75	96.38	97.97	96.92	98.71	97.47	98.55	97.49
Mean majority class strength (%)	98.48	97.60	98.57	97.82	99.09	98.23	99.01	98.24
Mean classification uncertainty	0.31	0.31	0.41	0.41	0.40	0.42	0.37	0.39

Note: “4oM” refers to “GPT-4o-mini,” while “3.5T” refers to “GPT-3.5-turbo.” See Appendix B for metric definitions.

Table 6 Summary Consistency Results for Summarization Tasks

Metric	MD&As (1)	Earnings Call Presentations (2)
Cosine-similarity metrics		
Mean run-pair similarity	0.98	0.98
Mean document-level similarity	0.98	0.98
Length metrics		
Run-level Metrics		
ICC2	0.66	0.75
Mean concordance correlation	0.43	0.56
Mean Pearson correlation	0.66	0.75
Mean Spearman correlation	0.66	0.75
Mean run-pair MARD (%)	8.51	8.37
Document-level Metrics		
Documents with length (%)	0.00	0.10
Mean document-wise MARD (%)	8.51	8.37
Tone Metrics		
Overall Inter-rater Agreements		
Fleiss' Kappa	0.68	0.65
Krippendorff's Alpha	0.68	0.65
Run-level Agreements		
Mean run-pair Cohen's Kappa	0.68	0.65
Mean run-pair agreement (%)	84.40	95.55
Document-level Agreements		
Percentage of perfect agreement (%)	48.10	82.20
Mean document-wise agreement (%)	84.40	95.55
Mean majority class strength (%)	89.03	96.92
Mean classification uncertainty	0.48	0.39

Note: See Appendix B for metric definitions.

Table 7 Summary Consistency Results for Text Generation Tasks

Metric	GPT-4o-mini (1)	GPT-3.5-turbo (2)
Cosine-similarity metrics		
Mean run-pair similarity	0.97	0.94
Mean document-level similarity	0.97	0.94
Length metrics		
Run-level Metrics		
ICC2	0.89	0.76
Mean concordance correlation	0.79	0.58
Mean Pearson correlation	0.89	0.76
Mean Spearman correlation	0.89	0.78
Mean run-pair MARD (%)	8.18	14.91
Document-level Metrics		
Documents with identical length (%)	0.05	0.10
Mean document-wise MARD (%)	8.18	14.91
Tone Metrics		
Overall Inter-rater Agreements		
Fleiss' Kappa	0.66	0.65
Krippendorff's Alpha	0.66	0.65
Run-level Agreements		
Mean run-pair Cohen's Kappa	0.66	0.65
Mean run-pair agreement (%)	90.64	85.34
Document-level Agreements		
Percentage of perfect agreement (%)	67.00	49.55
Mean document-wise agreement (%)	90.64	85.34
Mean majority class strength (%)	93.40	89.67
Mean classification uncertainty	0.45	0.46

Note: See Appendix B for metric definitions.

Table 8 Summary Consistency Results for Prediction Tasks

Metric	GPT-4o	GPT-4o-mini	GPT-3.5-turbo
Directional Prediction			
Overall Inter-rater Agreements			
Fleiss' Kappa	0.98	0.99	0.97
Krippendorff's Alpha	0.98	0.99	0.97
Run-level Agreements			
Mean run-pair Cohen's Kappa	0.98	0.99	0.97
Mean run-pair agreement (%)	99.19	99.33	98.54
Document-level Agreements			
Percentage of perfect agreement (%)	94.90	97.30	93.70
Mean document-wise agreement (%)	99.19	99.33	98.54
Mean majority class strength (%)	99.49	99.51	99.05
Mean classification uncertainty	0.27	0.39	0.37
Point Forecast			
Run-level Metrics			
ICC2	0.997	0.864	0.846
Mean concordance correlation	0.995	0.759	0.745
Mean Pearson correlation	0.997	0.868	0.878
Mean Spearman correlation	0.999	0.971	0.969
Mean run-pair MARD (%)	2.05	6.18	9.89
Document-level Metrics			
Documents with point estimate (%)	29.20	32.80	10.90
Mean document-wise MARD (%)	2.05	6.42	9.89

Note: See Appendix B for metric definitions.

Online Appendix A:
Task-Specific Experimental Design

In this appendix, we begin with a high-level overview of the task-specific experimental designs, including sample selection, model choice, run size, batch size, and API calling. The details for each task are elaborated in the corresponding sub-sections. Table A1 presents a breakdown of the tasks by model, showing the sample size, number of runs, batch size, and total API calls for each task-model combination. The batch size refers to the number of inputs (e.g., sentences or paragraphs) sent in each API call, which is determined based on computing efficiency and cost considerations, as the prompts are repeated and incur costs. For instance, classification tasks using MD&As with a sample size of 10,000 sentences are conducted using GPT-3.5-turbo and GPT-4o-mini models, with 50 runs and a batch size of 10, resulting in a total of 50,000 API calls per model (10,000 sentences × 50 runs ÷ 10 sentences per batch). Across all tasks and models, the study involves a total of approximately 750,000 API calls, which produce more than 3.4 million outputs.

A1 Binary Classification

We select a sample of 10-K filings spanning from 2018 to 2024 based on filing dates, with the latest 10-K filings included up to December 14, 2024, which marks the initiation of our data analysis process for this task. This period allows us to examine corporate disclosures in the relatively recent past while straddling the pre- and post-LLM eras.

We begin by merging firm-year observations from the Compustat-CRSP intersection with 10-K filings over the same period. From the matched 10-K filings, we programmatically extract MD&A sections. We require a minimum word count of 500 words for the MD&A section to ensure sufficient content for analysis. To create a balanced and representative sample, we employ a stratified sampling approach targeting 5,000 unique firms that are distributed relatively evenly over the sample period.

For our analysis of forward-looking statements (FLS), we tokenize each MD&A into sentences and randomly select two unique sentences from each MD&A, while ensuring that the sentences from all MD&As are also unique. To ensure that the sentences are complete and not headings, we require the sentences to have at least 100 characters and end with a period. This process results in a final sample of 10,000 unique sentences.

The prompt for this task is provided in Panel A of Figure A1. The prompt establishes the model as an experienced financial analyst with specialized knowledge in analyzing MD&As of 10-K filings. This design choice ensures that the model approaches the task with domain-specific context and expertise. The prompt provides explicit instructions for classifying statements as either forward-looking or non-forward-looking, requiring outputs in a standardized JSON format to facilitate subsequent parsing. For operational efficiency and cost optimization, we send texts in batches of 10 sentences per request. These sentences are provided through the user prompt, maintaining separation between the task instructions (system prompt) and the content for analysis (user prompt).

A2 Multi-class Classification

Our sample period spans from January 3, 2006 to December 7, 2024. The former date marks the first date that FOMC statements are available in HTML format on the Federal Reserve website, while the latter date represents the date when we began collecting the data. We gather all 162

FOMC statements regarding monetary policy during this period. We parse these statements and remove the last paragraph that describes the voting results. We then tokenize the plain text into sentences and remove duplicate ones. This process yields 1,096 unique sentences, comprising approximately 27,000 words.

Following Hansen and Kazinnik (2024), we ask our chosen models to classify each sentence into one of five categories: "Hawkish," "Mostly Hawkish," "Neutral," "Mostly Dovish," or "Dovish." The system prompt used for this task is provided in Panel B of Figure A1. In this prompt, we ask the model to play the role of an economist who is good at analyzing monetary policy based on FOMC statements. For cost-effectiveness and computing efficiency, we supply 10 sentences per API call.

A3 Sentiment Analysis of News Articles

For this task, we employ the Financial PhraseBank dataset introduced by Malo et al. (2013). This dataset comprises sentences extracted from a diverse range of news sources, covering both small and large companies across various industries. This dataset was meticulously annotated by a group of 16 experts with in-depth knowledge of finance. These annotators were tasked with assessing each sentence from the perspective of an investor, determining whether the news conveyed would have a positive, negative, or neutral impact on the corresponding company's stock price. To ensure reliability of annotations, each sentence was evaluated by 5 to 8 annotators, resulting in the creation of four distinct groups based on the level of agreement among the annotators. The agreement levels are 50%, 66%, 75%, and 100%, with the majority labels assigned accordingly. We use the full dataset, which consists of 4,838 sentences.

The system prompt used for this task is provided in Panel A of Figure A2. In this prompt, we ask the model to assume the role of an experienced financial analyst and classify each sentence as positive, negative, or neutral. For cost-effectiveness and computing efficiency, we similarly supply 10 sentences per API call.

A4 Sentiment Analysis of MD&As

For this task, we generate the sentences from MD&As following the same procedures as those for binary classification of forward-looking statements, as detailed in Section A1. The prompt is provided in Panel B of Figure A2. We similarly ask the model to play the role of an experienced financial analyst, and we use a batch size of 10 sentences to reduce the number of API calls.

A5 Sentiment Analysis of Earnings Calls

Our initial sample consists of quarterly earnings conference call transcripts from the beginning of 2018 to June 30, 2024. We collect the transcripts from SeekingAlpha. June 30, 2024 represents the last date for which we were able to collect the transcripts using a semi-manual procedure with relative ease. We match these transcripts with the intersection of Compustat and CRSP databases. To ensure that the transcripts contain sufficient content, we require each transcript to have a presentation section and a Q&A section with at least 1,000 characters each, and a minimum of three questions asked during the Q&A session.

For computational efficiency and economic considerations, we select a subsample of 1,000 transcripts from the initial sample. To ensure that the final sample provides a good representation of each industry (based on two-digit SIC codes) and a relatively even distribution across years, we employ a stratified sampling approach. Furthermore, we require each industry to have at least two transcripts in the final sample.

As the final input for sentiment analysis, we randomly select two paragraphs from the presentation section of each transcript, excluding paragraphs containing the boilerplate forward-looking statement, which companies tend to provide at the beginning of the call. We require each selected paragraph to have at least 200 characters. In rare cases where there are not enough paragraphs meeting this requirement, we relax the criterion to include paragraphs with at least 100 characters. This process results in a final sample of 2,000 paragraphs. We apply the same requirements to obtain 2,000 paragraphs from executives' answers to questions during the Q&A sessions.

The prompts for this task are included in Panel C and Panel D of Figure A2. We instruct the model to assume the role of an experienced financial analyst and use a batch size of 10 paragraphs for processing the data.

A6 Summarization of MD&As

We begin, as with the binary classification task described in Section A1, by collecting 10-Ks filed between 2018 and 2024 from the Compustat-CRSP intersection. From these filings, we extract MD&A sections, requiring each section to contain a minimum of 500 words.

To ensure computational efficiency and economic feasibility, we construct a stratified random sample of 1,000 MD&As from this initial dataset. We employ a two-dimensional stratification approach to maintain representativeness. First, we select only one MD&A per unique company to avoid firm-level clustering. Second, we balance the sample across both temporal dimensions (filing years) and firm size (market value quartiles). The latter is particularly important given its association with disclosure complexity (Li 2008, Loughran and McDonald 2014). To accommodate the context window constraints of the language model, we exclude MD&As exceeding 50,000 words, which affects only 15 cases out of our initial sample of over 25,000 MD&As.

The prompt, provided in Panel A of Figure A3, instructs the language model to assume the role of an assistant to a financial analyst who follows a public company. The model's task is to help summarize the MD&A section of the company's 10-K report. This approach allows the language model to focus on the most relevant information within the MD&A without being overloaded by excessive context. By framing the task as an assistant to a financial analyst, we encourage the model to prioritize the extraction of key insights and metrics that are most valuable to investors and analysts. To ensure that the input does not exceed the model's context window, we include only one MD&A section per API call.

A7 Summarization of Earnings Call Presentations

We generate a sample of 1,000 earnings call transcripts using the same method described in Section A5. We then extract the presentation sections of the earnings call transcripts for summarization. We choose the presentation sections because they tend to be lengthy and would benefit from summarization. This approach mimics practical applications of using a large language model to aid decision making. The prompt is provided in Panel B of Figure A3. We send one presentation for each API call to ensure that the context window is not exceeded.

A8 Text Generation of Answering Earnings Call Questions

We generate a sample of 1,000 transcripts as described in Section A5. We then randomly select two questions posed by participating financial analysts from the Q&A section of each transcript. In addition to the questions, we provide context information to allow the model to answer them. Following Bai et al. (2023), we provide a summary of the presentation section and the previous questions and their answers up to the current question.

The prompt is provided in Figure A4. In this prompt, we ask the model to assume the role of an executive from the company who needs to answer questions from financial analysts during the Q&A section. We instruct the model to answer the questions solely based on the context information provided. In cases where the question is the first question, the model should answer based only on the summary of the presentation.

We generate the summaries through a separate procedure using GPT-4o-mini and a slightly different prompt. The prompt is the same as that provided in Panel B of Figure A3, except that we ask the model to assume a different role by including the following statement:

```
You are an assistant to an executive of a public company. Your task
is to summarize the presentation section of an earnings conference
call transcript. The executive will rely on this summary to answer
a question from a financial analyst following the company.
```

We generate the summaries only once, and for each and every run, we use the same summary. This approach allows us to assess whether the model is able to produce consistent answers across different runs, given the same information.

A9 Prediction of Future Earnings

Our sample selection for this task starts with the intersection of Compustat, CRSP, and EDGAR. The initial sample period is from January 1, 2023, to October 30, 2024, based on fiscal year end. The latter date marks the most recent data date available in Compustat when we started to process the data. We require firms to have positive sales, total assets, and book equity. We remove financial and utility firms, which have different financial statement formats or earnings patterns.

To avoid lookahead bias that tends to plague prediction tasks (Dong et al. 2024, Levy 2024), we focus on firms with earnings announcement dates on or after November 1, 2023, because two models we use for this task have a knowledge cut-off of October 2023. We require firm-years to have financial data for the prediction year and the previous three years.

To generate a representative sample, we randomly choose 1,000 unique companies across four market value quartiles and ensure that 50% of them have increased earnings and 50% have decreased earnings based on the earnings of the prediction year. Requiring data availability for the prediction year helps us create a balanced sample in terms of earnings changes. We construct the condensed balance sheet and financial statements, largely follow Li et al. (2024) and Kim et al. (2024). For both statements, we provide the data for three years, namely, the current year (t) and the previous two years (t-1, and t-2), as is customary with comparative statements provided in annual reports. We ask the model to predict the net income before extraordinary items for the following year, i.e., year t+1.

In addition to the financial statements, we provide a summary of the current year's MD&A from the 10-K filing. This mimics the fact that when investors or other professionals, such as financial analysts, predict future earnings, they can rely on corporate disclosures. We generate the MD&A

summaries through a separate procedure, using a slightly different prompt from that provided in Panel A of Figure A3, by including the following statement so that the model understands its role and the purpose of the task:

> You are the assistant to an experienced financial analyst with expertise in predicting future earnings based on company disclosures. Your task is to summarize the Management’s Discussion and Analysis (MD&A) section from a 10-K filing. The financial analyst will use this summary along with comparative financial statements to forecast the company's earnings for the next period.

Based on the MD&A summary and comparative statements, we ask the model to first predict whether net income before extraordinary items will increase or decrease in the next fiscal year, and we also ask the model to provide a point estimate of the value. We provide the detailed prompt in Figure A5. We also provide the full user prompt, which includes the summary and statements. Note that we have formatted the statements so that they are more readable to human readers. When we provide the data to the model, we provide them in a markdown format so that the model can parse them more easily. For example, the heading of a statement will be shown as "| Line Items | Current Year (t) | Year (t-1) | Year (t-2) |" in a markdown format, and a line item will be shown as “| Net sales | 1,990.600 | 1,817.100 | 1,651.400 |”.

We disclose the company name in the heading of the MD&A summary. We believe this information is important for the model as it allows the model to use its existing knowledge about the company. This mimics the situation when human forecasters are asked to do the job, as they have pre-existing knowledge of the company.

Figure A1 Prompts for Classification Tasks

Panel A: Binary classification of Forward-looking Statements from MD&As

```
You are an experienced financial analyst with expertise in analyzing
Management Discussion & Analysis (MD&A) sections of 10-K filings. Your task
is to classify sentences based on whether they contain forward-looking
information.

Guidelines:
1. Analyze each sentence individually.
2. Return your analysis as a JSON object where:
   - The key is the provided sentence ID (e.g., "#1").
   - The value should be either ["Yes"] for forward-looking statements or
["No"] for non-forward-looking statements.

Output Requirements:
Output the entire JSON object as a single line without extra spaces or line
breaks.

Example Output:
{"#1":["Yes"],"#2":["No"],"#3":["Yes"]}

Here are the sentences to analyze:
                           [A list of 10 sentences]
```

Panel B: Multi-class Classification of Sentences from FOMC Statements

```
You are an economist specializing in monetary policy analysis. Your task is
to classify each sentence from FOMC statements into one of five categories:
Hawkish, Mostly Hawkish, Neutral, Mostly Dovish, or Dovish.

Guidelines:
1. Analyze each sentence individually.
2. Return your analysis as a JSON object where:
   - The key is the provided sentence ID (e.g., "#1").
   - The value is one of the following options: ["Hawkish"], ["Mostly
Hawkish"], ["Neutral"], ["Mostly Dovish"], or ["Dovish"].

Output Requirements:
Output the entire JSON object as a single line without extra spaces or line
breaks.

Example Output:
{"#1":["Hawkish"],"#2":["Mostly Dovish"],"#3":["Neutral"],"#4":["Mostly
Hawkish"],"#5":["Dovish"]}

Here are the sentences to analyze:
                           [A list of 10 sentences]
```

Figure A2 Prompt for Sentiment Analysis Tasks

Panel A: Financial News

```
You are an experienced financial analyst with expertise in analyzing news
sentiment. Your task is to classify the tone of each sentence about companies
as Positive, Negative, or Neutral.

Guidelines:
1. Analyze each sentence individually.
2. Return your analysis as a JSON object where:
   - The key is the provided sentence ID (e.g., "#1").
   - The value is one of the following options: ["Positive"], ["Negative"],
or ["Neutral"].

Output Requirements:
Output the entire JSON object as a single line without extra spaces or line
breaks.

Example Output:
{"#1":["Positive"],"#2":["Negative"],"#3":["Neutral"]}

Here are the sentences to analyze:
                           [A list of 10 sentences]
```

Panel B: MD&A Sentences

```
You are an experienced financial analyst with expertise in analyzing
Management Discussion & Analysis (MD&A) sections of 10-K filings. Your task
is to classify the sentiment of sentences from MD&A sections as Positive,
Negative, or Neutral.

Guidelines:
1. Analyze each sentence individually.
2. Return your analysis as a JSON object where:
   - The key is the provided sentence ID (e.g., "#1").
   - The value is one of the following options: ["Positive"], ["Negative"],
or ["Neutral"].

Output Requirements:
Output the entire JSON object as a single line without extra spaces or line
breaks.

Example Output:
{"#1":["Positive"],"#2":["Negative"],"#3":["Neutral"]}

Here are the sentences to analyze:
                           [A list of 10 sentences]
```

Appendix A2 Prompt for Sentiment Analysis Tasks (Continued)

Panel C: Earnings Conference Calls (Presentations)

```
You are an experienced financial analyst with expertise in analyzing earnings
conference call transcripts. Your task is to classify the sentiment of each
paragraph from earnings call presentation sections as Positive, Negative, or
Neutral.

Guidelines:
1. Analyze each paragraph individually.
2. Return your analysis as a JSON object where:
   - The key is the provided paragraph ID (e.g., "#1").
   - The value is one of the following options: ["Positive"], ["Negative"],
or ["Neutral"].

Output Requirements:
Output the entire JSON object as a single line without extra spaces or line
breaks.

Example Output:
{"#1":["Positive"],"#2":["Negative"],"#3":["Neutral"]}

Here are the paragraphs to analyze:
                           [A list of 10 paragraphs]
```

Panel D: Earnings Conference Calls (Q&As)

```
You are an experienced financial analyst with expertise in analyzing earnings
conference call transcripts. Your task is to classify the sentiment of each
paragraph from earnings call Q&A sections as Positive, Negative, or Neutral.

Guidelines:
1. Analyze each paragraph individually.
2. Return your analysis as a JSON object where:
   - The key is the provided paragraph ID (e.g., "#1").
   - The value is one of the following options: ["Positive"], ["Negative"],
or ["Neutral"].

Output Requirements:
Output the entire JSON object as a single line without extra spaces or line
breaks.

Example Output:
{"#1":["Positive"],"#2":["Negative"],"#3":["Neutral"]}

Here are the paragraphs to analyze:
                           [A list of 10 paragraphs]
```

Figure A3 Prompts for Summarization Tasks

Panel A: MD&As

```
You are an assistant to a financial analyst who follows a public company.
Your task is to summarize the Management’s Discussion and Analysis (MD&A)
section from a 10-K filing of the company.

Return your analysis as a JSON object where:
  - The key is the provided document ID (e.g., "#1").
  - The value is a summary of the MD&A.

Output Requirements:
Output the entire JSON object as a single line without extra spaces or line
breaks.

Example Output:
{"#1": ["Summary of MD&A"]}

MD&A Section to Summarize:
                              [One MD&A section]
```

Panel B: Earnings Conference Call Presentations

```
You are an assistant to a financial analyst who follows a public company.
Your task is to summarize the presentation section of the company's earnings
conference call transcript for the analyst.

Return your summary as a JSON object where:
  - The key is the provided document ID (e.g., "#1").
  - The value is the summary of the presentation section.

Output Requirements:
Output the entire JSON object as a single line without extra spaces or line
breaks.

Example Output:
{"#1": ["Summary of the presentation section"]}

Presentation section to summarize:
[One presentation section]
```

Figure A4 Prompt for Text Generation Task

```
You are an executive of a public company participating in an earnings
conference call. Your role is to respond to a financial analyst's question
during the Q&A session.

Instructions:
1. Use the Presentation Summary and any Previous Q&A provided to guide your
response.
2. If this is the first question, craft your response based solely on the
Presentation Summary.
3. Input Format:

Document ID: A unique identifier for the reference material (e.g., #1).
Presentation Summary: Key points from the earnings call presentation.
Previous Q&A (if any): Questions from analysts and responses provided
earlier.
Current Question: The analyst's question that you need to address.

Output Format:
1. Return your response as a JSON object:
 - Key: The Document ID of the reference material (e.g., #1).
 - Value: Your response to the analyst's question.
2. Ensure the JSON object is formatted on a single line without extra spaces
or line breaks.

Example Output:
{"#1": ["Your answer to the question."]}

Reference Material:
                     [Presentation Summary + Previous Q&A]
```

Figure A5 Prompt for Prediction Task

Panel A: System Prompt for Earnings Prediction

```
You are an experienced financial analyst. Your task is to forecast the income
before extraordinary items of a public company for the next fiscal year. Your
analysis should be based on the following data:

1. A summary of Management's Discussion and Analysis (MD&A) from the current
year's 10-K filing
2. Comparative balance sheets for the current and previous two years
3. Comparative income statements for the current and previous two years

Provide your forecast in two parts:
1. Direction: Whether income before extraordinary items will "Increase" or
"Decrease"
2. Amount: A point estimate using the same unit as the income statement (do
not include the unit)

Return your forecast as a JSON object:
  - Key: The document ID as provided (e.g., "#1")
  - Value: An array containing the direction and point estimate (e.g.,
["Increase", "456.326"])

Format Requirements:
  - Output the JSON object on a single line without extra spaces or line
breaks.
  - Use exactly three decimal places for the point estimate.

Example output: {"#1": ["Increase", "638.463"]}

Please analyze the following data for your forecast:

        [MD&A Summary + Comparative balance sheet and Income statement]
```

Figure A5 Prompt for Prediction Task (Continued)

Panel B: Sample User Prompt for Earnings Prediction

MD&A summary of the current year for AAR CORP:

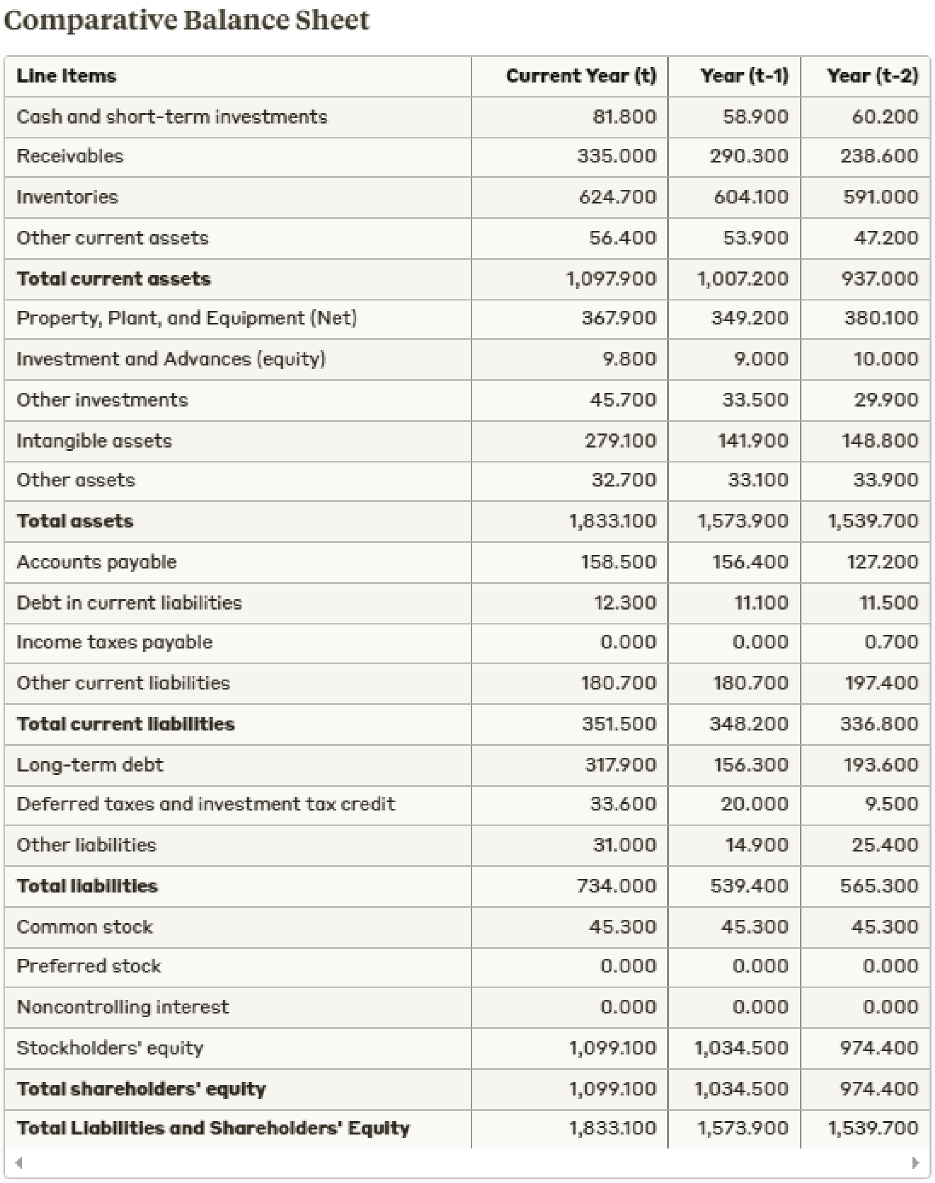
The MD&A discusses the company's financial condition and results of operations, emphasizing the importance of reading it alongside the consolidated financial statements. It outlines the company's two business segments: Aviation Services and Expeditionary Services, detailing their operations and revenue sources. The company has restructured its Aviation Services segment into three new operating segments for better performance evaluation. Fiscal 2023 saw a recovery in air travel, leading to a 22.6% increase in sales to commercial customers, while sales to government customers decreased by 10.1%. The company secured new long-term contracts and made a significant acquisition of Trax USA Corp. for $120 million, enhancing its software offerings. Operating cash flows were $23.8 million, down from $89.8 million in the previous year, primarily due to increased investments. The company maintains a strong liquidity position with $746.4 million in working capital. The MD&A also highlights the impact of rising interest rates on expenses and outlines the company's strategies for future growth, including investments in both commercial and government markets.

[Comparative Balance Sheet in Markdown Format – A sample is provided in Panel C]

[Comparative Balance Sheet in Markdown Format – A sample is provided in Panel D]

Figure A5 Prompt for Prediction Task (Continued)

Panel C: Sample Comparative Balance Sheet

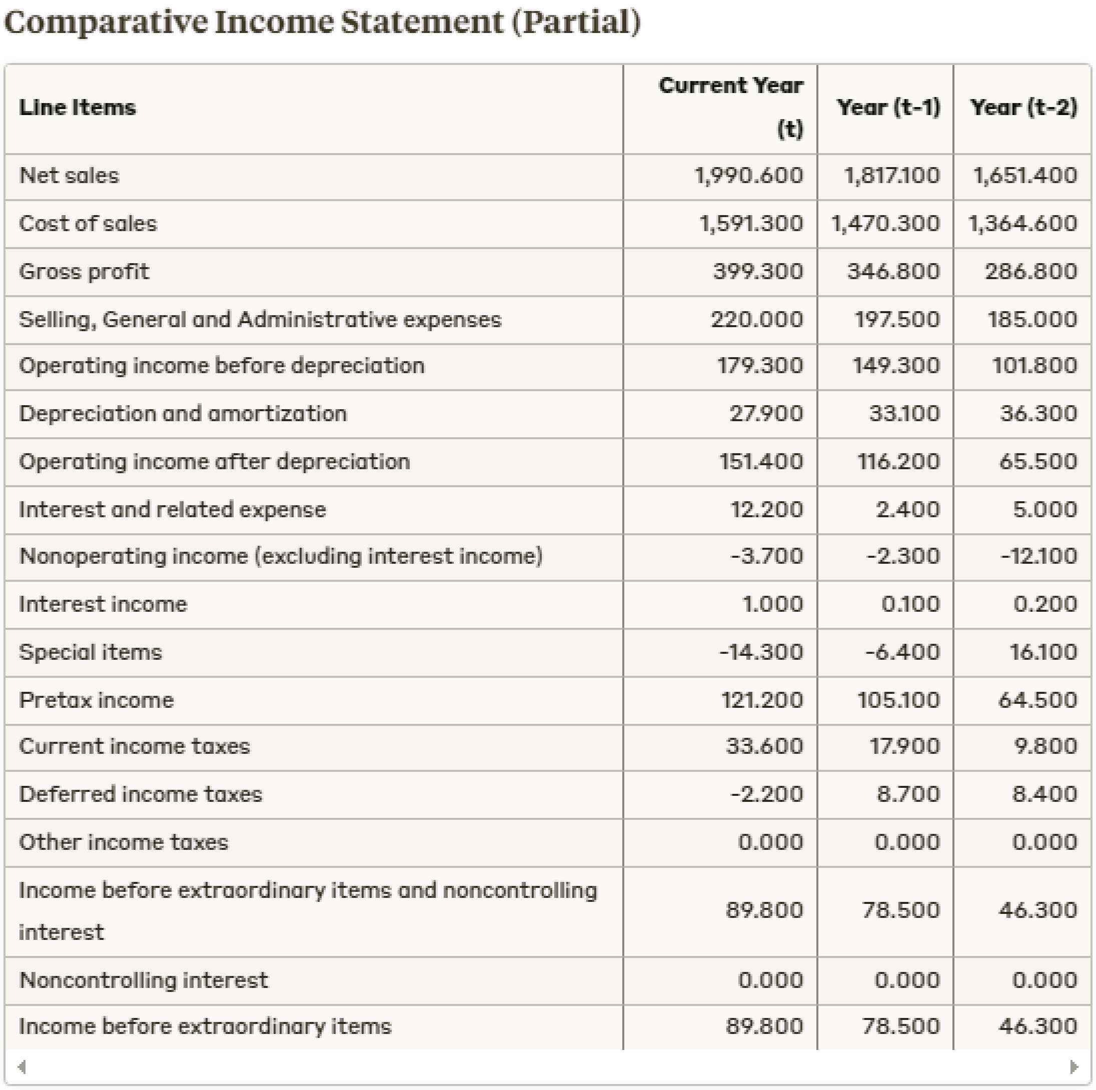


Comparative Balance Sheet

Line Items	Current Year (t)	Year (t-1)	Year (t-2)
Cash and short-term investments	81.800	58.900	60.200
Receivables	335.000	290.300	238.600
Inventories	624.700	604.100	591.000
Other current assets	56.400	53.900	47.200
Total current assets	1,097.900	1,007.200	937.000
Property, Plant, and Equipment (Net)	367.900	349.200	380.100
Investment and Advances (equity)	9.800	9.000	10.000
Other investments	45.700	33.500	29.900
Intangible assets	279.100	141.900	148.800
Other assets	32.700	33.100	33.900
Total assets	1,833.100	1,573.900	1,539.700
Accounts payable	158.500	156.400	127.200
Debt in current liabilities	12.300	11.100	11.500
Income taxes payable	0.000	0.000	0.700
Other current liabilities	180.700	180.700	197.400
Total current liabilities	351.500	348.200	336.800
Long-term debt	317.900	156.300	193.600
Deferred taxes and investment tax credit	33.600	20.000	9.500
Other liabilities	31.000	14.900	25.400
Total liabilities	734.000	539.400	565.300
Common stock	45.300	45.300	45.300
Preferred stock	0.000	0.000	0.000
Noncontrolling interest	0.000	0.000	0.000
Stockholders' equity	1,099.100	1,034.500	974.400
Total shareholders' equity	1,099.100	1,034.500	974.400
Total Liabilities and Shareholders' Equity	1,833.100	1,573.900	1,539.700

Figure A5 Prompt for Prediction Task (Continued)

Panel D: Sample Comparative Income Statement



Comparative Income Statement (Partial)

Line Items	Current Year (t)	Year (t-1)	Year (t-2)
Net sales	1,990.600	1,817.100	1,651.400
Cost of sales	1,591.300	1,470.300	1,364.600
Gross profit	399.300	346.800	286.800
Selling, General and Administrative expenses	220.000	197.500	185.000
Operating income before depreciation	179.300	149.300	101.800
Depreciation and amortization	27.900	33.100	36.300
Operating income after depreciation	151.400	116.200	65.500
Interest and related expense	12.200	2.400	5.000
Nonoperating income (excluding interest income)	-3.700	-2.300	-12.100
Interest income	1.000	0.100	0.200
Special items	-14.300	-6.400	16.100
Pretax income	121.200	105.100	64.500
Current income taxes	33.600	17.900	9.800
Deferred income taxes	-2.200	8.700	8.400
Other income taxes	0.000	0.000	0.000
Income before extraordinary items and noncontrolling interest	89.800	78.500	46.300
Noncontrolling interest	0.000	0.000	0.000
Income before extraordinary items	89.800	78.500	46.300

Table A1 Summary of Task-Specific Experimental Designs

	Sample				**Batch**	**API**
Task	**size**	**Model**	**Runs**	**Size*Run**	**size**	**Calls**
Classification (MD&As)	10,000	GPT-3.5-turbo	50	500,000	10	50,000
Classification (MD&As)	10,000	GPT-4o-mini	50	500,000	10	50,000
Classification (FOMC)	1,096	GPT-3.5-turbo	50	54,800	10	5,480
Classification (FOMC)	1,096	GPT-4o-mini	50	54,800	10	5,480
Sentiment analysis (News)	4,838	GPT-3.5-turbo	50	241,900	10	24,190
Sentiment analysis (News)	4,838	GPT-4o-mini	50	241,900	10	24,190
Sentiment analysis (Call presentations)	2,000	GPT-3.5-turbo	50	100,000	10	10,000
Sentiment analysis (Call presentations)	2,000	GPT-4o-mini	50	100,000	10	10,000
Sentiment analysis (Call Q&As)	2,000	GPT-3.5-turbo	50	100,000	10	10,000
Sentiment analysis (Call Q&As)	2,000	GPT-4o-mini	50	100,000	10	10,000
Sentiment analysis (MD&As)	10,000	GPT-3.5-turbo	50	500,000	10	50,000
Sentiment analysis (MD&As)	10,000	GPT-4o-mini	50	500,000	10	50,000
Summarization (Call presentations)	1,000	GPT-4o-mini	50	50,000	1	50,000
Summarization (MD&As)	1,000	GPT-4o-mini	50	50,000	1	50,000
Text generation (Call answers)	2,000	GPT-3.5-turbo	50	100,000	1	100,000
Text generation (Call answers)	2,000	GPT-4o-mini	50	100,000	1	100,000
Prediction (future earnings)	1,000	GPT-3.5-turbo	50	50,000	1	50,000
Prediction (future earnings)	1,000	GPT-4o-mini	50	50,000	1	50,000
Prediction (future earnings)	1,000	GPT-4o	50	50,000	1	50,000
			Total	**3,443,400**		**749,340**

Online Appendix B:
Detailed Consistency Results

B1 Binary Classification of Forward-looking Statements

Table B1 presents the consistency analysis results for the binary classification of forward-looking statements from MD&As. Our analysis encompasses a sample of 10,000 sentences, each classified 50 times by two different models: GPT-3.5-turbo and GPT-4o-mini. While we use sentences in this task, we adopt the general term "document" to refer to individual text units, which may be paragraphs or even full MD&A sections for some tasks.

B1.1 Inter-rater Agreement

Both models demonstrate exceptionally high consistency and reproducibility. The overall inter-rater agreement metrics—Fleiss' Kappa and Krippendorff's Alpha—yield values of 0.93 for GPT-3.5-turbo and 0.97 for GPT-4o-mini, as shown in Panel A of Table B1. These two metrics appear identical to each other at two decimal places (though they differ at higher precision), because both of them measure inter-rater consistency by accounting for chance agreement, albeit through different mathematical approaches. According to standard interpretations, a Krippendorff's Alpha above 0.8 indicates high consistency (Krippendorff 2004).

B1.2 Run-level Agreement

At the run level, complementary perspectives on consistency are provided by the mean run-pair Cohen's Kappa score and run-pair agreement. The mean Cohen's Kappa score, calculated across all 1,225 possible run pairs ((50 × 49)/2), aligns with the overall inter-rater agreement metrics. Both models maintain high consistency across different runs, with GPT-3.5-turbo achieving 98.07% run-pair agreement and GPT-4o-mini reaching 99.00%. T-tests confirm that this difference of 0.93 percentage points is statistically significant at the 0.001 level ($p < 0.01$). A practical interpretation of a 99% run-pair agreement is that, given two independent runs, 99% of documents will receive the same classification across those two runs.

B1.3 Document-level Agreement

The document-level agreements offer a more granular view of consistency. Both models achieve impressive perfect agreement rates (identical classification across all 50 runs): 92.17% for GPT-3.5-turbo and 96.18% for GPT-4o-mini, representing a substantial 4.01 percentage point difference. The mean document-wise agreement (98.07% vs. 99.00%) and majority class strength metrics (98.67% vs. 99.30%) are notably high for both models, with GPT-4o-mini showing significantly better performance in both metrics ($p < 0.01$) based on t-tests. The mean classification uncertainty remains low for both models (0.38 for GPT-3.5-turbo and 0.40 for GPT-4o-mini), with this small difference (0.02) not reaching statistical significance based on independent t-test results. Notably, the identical values for mean run-pair agreement and mean document-wise agreement are expected, as both metrics are calculated from the same underlying pairwise comparisons, differing only in their level of aggregation.

B1.4 Class-specific Performance

Both models show higher consistency in identifying non-forward-looking statements (98-99%) than forward-looking ones (94-97%), suggesting the latter are more challenging to classify. GPT-4o-mini identifies significantly more forward-looking statements (24.30%) than GPT-3.5-turbo

(17.76%), which likely indicates improved accuracy in detecting subtle forward-looking language in financial texts, since GPT-4o-mini is a newer and more advanced model.

B1.5 Detailed Run-level Analysis

Panel B of Table B1 reveals strong run-level agreement statistics for both models. For GPT-3.5-turbo, Kappa scores across all 1,225 run pairs average 0.93 (SD=0.01, range: 0.90-0.95), with mean run-pair agreement at 98.07% (SD=0.20%, IQR: 97.98%-98.19%). For GPT-4o-mini, performance is superior, showing a mean Kappa of 0.97 (SD=0.00) and run-pair agreement of 99.00% (SD=0.09%, IQR: 98.93%-99.06%). This tight distribution demonstrates exceptional stability—practically speaking, two independent users running GPT-4o-mini on 10,000 documents would receive identical classifications for approximately 9,900 documents.

B1.6 Document-level Uncertainty

Panel C shows both models achieve high document-level consistency. For GPT-3.5-turbo, while mean document-wise agreement is 98.07%, the median of 100% indicates most documents have complete agreement across 50 runs. Among the 7.83% of documents with any disagreement, the mean uncertainty score is 0.38, suggesting relatively decisive classifications even in contested cases.

GPT-4o-mini demonstrates superior consistency with 99.00% mean document-wise agreement and only 3.82% of documents receiving different classifications across runs—half the amount observed with GPT-3.5-turbo. When GPT-4o-mini does show uncertainty, its pattern (mean=0.40) resembles GPT-3.5-turbo's decisiveness in disagreements.

To better understand the uncertainty characteristics, we plot the distribution of the uncertainty scores for both models in Figure B1. The plot shows that when disagreements do occur, both models exhibit similar uncertainty distributions, with slight peaks at both low (0.1) and high (0.7) entropy scores. This bimodal tendency suggests that disagreements tend to be either highly decisive (low entropy) or highly contentious (high entropy). Importantly, GPT-4o-mini has significantly fewer documents with any disagreements (382 vs. 783), further confirming its superior consistency.

B1.7 Implications

Both GPT-3.5-turbo and GPT-4o-mini demonstrate remarkably high consistency and reproducibility in this classification task, with agreement metrics well above conventional thresholds for "almost perfect" agreement. This indicates that both models can provide stable and reliable classifications for financial text analysis. Within this context of strong overall performance, GPT-4o-mini shows consistently superior reliability across all agreement metrics, with statistically significant improvements in most cases.

The higher proportion of identified forward-looking statements by GPT-4o-mini (24.30% versus 17.76%) raises important considerations for financial text analysis applications. This discrepancy warrants further investigation in future research, particularly to establish whether GPT-4o-mini's increased detection rate represents an improvement in accuracy or potentially a different classification boundary.

B2 Multi-class Classification of FOMC Statements

Table B2 presents the consistency analysis results for the multi-class classification of FOMC statements. Our analysis examines 1,096 sentences classified 50 times by two models: GPT-3.5-turbo and GPT-4o-mini. Unlike the binary classification task, this task involves categorizing statements into five distinct classes (Hawkish, Mostly Hawkish, Neutral, Mostly Dovish, and Dovish), representing a more complex classification challenge.

B2.1 Inter-rater Agreement

Both models demonstrate good consistency, though notably lower than in the binary classification task. The Fleiss' Kappa and Krippendorff's Alpha values are 0.91 for GPT-3.5-turbo and 0.86 for GPT-4o-mini, as shown in Panel A of Table B2. Unlike the binary classification results, GPT-3.5-turbo shows higher consistency in this multi-class task, with a difference of 0.05 in both metrics.

B2.2 Run-level Agreement

The run-level metrics confirm this pattern, with GPT-3.5-turbo achieving a mean Kappa score of 0.91 and run-pair agreement of 92.96%, compared to GPT-4o-mini's 0.86 and 88.93%, respectively. These differences of 0.05 in Kappa and 4.03 percentage points in run-pair agreement are statistically significant ($p < 0.01$).

B2.3 Document-level Agreement

At the document level, GPT-3.5-turbo again demonstrates superior consistency with 65.33% perfect agreement across all 50 runs compared to GPT-4o-mini's 55.66%, a substantial 9.67 percentage point difference. The mean document-wise agreement (92.96% vs. 88.93%) and majority class strength (95.32% vs. 92.26%) are significantly higher for GPT-3.5-turbo ($p < 0.01$). The mean classification uncertainty is lower for GPT-3.5-turbo (0.34 vs. 0.40), with this difference of 0.06 being statistically significant.

B2.4 Class-specific Performance

Class-specific agreement metrics reveal that both models struggle more with intermediate categories than with extreme ones. For GPT-3.5-turbo, the highest agreements are for "Neutral" (93.66%) and "Hawkish" (93.13%), while "Mostly Dovish" (91.34%) and "Dovish" (90.22%) show lower consistency. For GPT-4o-mini, "Neutral" statements also achieve the highest agreement (91.83%), while "Mostly Hawkish" statements have the lowest (87.19%). Across all classes, GPT-3.5-turbo consistently outperforms GPT-4o-mini, with differences ranging from 1.61 to 6.44 percentage points.

B2.5 Class Distribution Differences

The class distribution shows substantial differences between the models. GPT-4o-mini classifies significantly more statements as "Dovish" (21.35% vs. 10.95%, +10.40 percentage points) and "Hawkish" (8.30% vs. 6.02%, +2.28 percentage points), while GPT-3.5-turbo assigns more statements to intermediate categories, particularly "Mostly Dovish" (25.36% vs. 19.07%, +6.30 percentage points) and "Mostly Hawkish" (23.36% vs. 18.70%, +4.65 percentage points). Both models classify approximately one-third of statements as "Neutral" (34.31% for GPT-3.5-turbo vs. 32.57% for GPT-4o-mini).

B2.6 Detailed Run-level Analysis

Panel B provides detailed statistics for run-level agreements. For GPT-3.5-turbo, Kappa scores show strong consistency with a mean of 0.91 (standard deviation = 0.06), ranging from 0.69 to 0.96. The run-pair agreement averages 92.96% (standard deviation = 4.28%), with the 25th and 75th percentiles at 93.52% and 94.62%, respectively.

For GPT-4o-mini, metrics indicate lower consistency, with a mean Kappa score of 0.86 (standard deviation = 0.06) and mean run-pair agreement of 88.93% (standard deviation = 4.51%). The minimum agreement drops to 71.81%, significantly lower than GPT-3.5-turbo's minimum of 77.28%.

B2.7 Document-level Analysis

Panel C reveals that while both models achieve high document-level consistency, GPT-3.5-turbo's performance is superior. Despite median document-wise agreement of 100% for both models, GPT-3.5-turbo's mean agreement of 92.96% exceeds GPT-4o-mini's 88.93%. Among documents with disagreements, GPT-3.5-turbo shows lower uncertainty (0.34) compared to GPT-4o-mini (0.40), suggesting more decisive classifications even in contested cases.

B2.8 Key Differences from Binary Classification

Unlike the binary classification task where GPT-4o-mini demonstrated superior consistency, in this multi-class task, GPT-3.5-turbo shows consistently better performance across all agreement metrics. This reversal suggests that the newer GPT-4o-mini model may be more sensitive to nuanced distinctions between closely related categories (e.g., "Hawkish" vs. "Mostly Hawkish"), resulting in lower inter-rater agreement.

The substantial differences in class distribution—particularly GPT-4o-mini's tendency to classify more statements in extreme categories while GPT-3.5-turbo favors intermediate ones—highlights important considerations for financial text analysis applications. This pattern suggests fundamentally different classification boundaries between the models, with GPT-4o-mini potentially exhibiting a more decisive stance in monetary policy classification.

B2.9 Implications

Both models demonstrate good consistency for this complex five-class task, though neither reaches the exceptional levels observed in the binary classification. The decreased agreement metrics across all measures compared to the binary task reflect the inherently greater challenge of distinguishing between five closely related categories versus two distinct ones.

For applications requiring multi-class classification of FOMC statements, GPT-3.5-turbo appears to be the more reliable choice based on consistently higher agreement metrics. However, researchers should carefully consider the substantial differences in class distribution between models, as this could significantly impact downstream analyses and interpretations of monetary policy stance.

B3 Sentiment Analysis

In this section, we present the results for sentiment analysis across four diverse financial text sources: news articles, MD&A sections, earnings call presentations, and earnings call Q&As. Table B3 provides a comprehensive view of model consistency for sentiment classification tasks.

B3.1 Inter-rater Agreement

As shown in Panel A, both models demonstrate exceptionally high consistency across all text types. GPT-4o-mini consistently outperforms GPT-3.5-turbo, with Fleiss' Kappa and Krippendorff's Alpha values ranging from 0.96-0.98 for GPT-4o-mini compared to 0.94-0.96 for GPT-3.5-turbo. The highest agreement metrics are observed for earnings call presentations (0.98 for GPT-4o-mini), while news articles show the lowest, though still impressive, scores (0.94 for GPT-3.5-turbo).

B3.2 Run-level Agreement

The run-level metrics confirm a pattern of high consistency. Mean run-pair agreement exceeds 96% for all text types and models, with GPT-4o-mini achieving up to 98.71% for earnings call presentations. The difference between models is most pronounced for news articles (97.75% vs. 96.38%) and least notable for MD&A sections (97.97% vs. 96.92%). In untabulated statistical tests, these differences are significant across all text types.

B3.3 Document-level Agreement

At the document level, both models maintain impressive consistency. The percentage of perfect agreement (identical classification across all 50 runs) ranges from 81.09% (GPT-3.5-turbo on news) to 95.05% (GPT-4o-mini on presentations). GPT-4o-mini shows superior perfect agreement rates across all text types, with the largest advantage in news articles (88.28% vs. 81.09%). Majority class strength exceeds 97% for all combinations, indicating highly decisive classifications.

B3.4 Class-specific Performance

Both models demonstrate high agreement rates across all sentiment classes, with slightly higher consistency for negative sentiment classifications (exceeding 96% except for Q&A with GPT-3.5-turbo). Notably, neutral class agreement is consistently the lowest among the three sentiment categories, though still above 95%, suggesting greater ambiguity in classifying neutral content.

B3.5 Class Distribution Differences

The class distribution reveals significant differences across text types and between models. Earnings call presentations contain the highest proportion of positive statements (56.65-64.15%), while MD&A sections show the highest proportion of neutral content (51.38-55.93%). GPT-4o-mini consistently identifies more positive content than GPT-3.5-turbo across all text types, with the largest difference in Q&A sections (55.20% vs. 40.50%). This systematic difference suggests potentially different classification thresholds between the models.

B3.6 Detailed Analysis

Panels B and C provide detailed statistics on run-level and document-level agreements. Panel B reveals remarkably narrow distributions of Kappa scores and run-pair agreements, particularly for

GPT-4o-mini, which shows standard deviations as low as 0.11 percentage points for MD&A run-pair agreement. Even minimum agreement values remain high, with the lowest being 88.07% for news articles with GPT-3.5-turbo.

Panel C's document-level analysis confirms that both models maintain high consistency, with median document-wise agreement of 100% across all text types. The number of documents with any disagreement varies significantly, from just 99 (4.95%) for earnings call presentations with GPT-4o-mini to 915 (18.91%) for news articles with GPT-3.5-turbo. Classification uncertainty remains low across all text types and models, indicating decisive classifications even in contested cases.

B3.7 Implications

Both models demonstrate exceptional consistency in sentiment analysis across diverse financial text types, with agreement metrics well above conventional thresholds for "almost perfect" agreement. GPT-4o-mini shows consistently superior reliability across all agreement metrics and text types. The systematic differences in sentiment distribution—particularly GPT-4o-mini's tendency to classify more content as positive—warrant consideration when selecting models for financial sentiment analysis applications.

These findings suggest that while both models provide highly reliable sentiment classifications, researchers should be aware of the potential impact of model choice on the distribution of sentiment labels, which could significantly affect downstream financial analyses and interpretations.

B4 Summarization

B4.1 Semantic Similarity

For this task, we use GPT-4o-mini to summarize MD&As from 10-Ks and presentation sections of earnings call transcripts. Since the primary purpose of summarization is to provide condensed versions of longer texts that allow readers to quickly grasp essential content, the most critical aspect of consistency in summaries is semantic similarity. We assess this consistency using cosine similarity, a widely accepted measure in accounting and finance literature. We calculate the cosine similarity based on context-aware embeddings generated by "jina-embeddings-v3," a state-of-the-art embedding model that excels at semantic matching tasks.

Table B4 presents the results of our semantic similarity analysis. Overall, we observe remarkably high consistency in the summaries generated by GPT-4o-mini. For both MD&A sections and earnings call presentations, the mean similarity scores are 0.98 across both run-pair and document-level measurements. This indicates strong semantic consistency between different runs of the model on the same input texts.

The run-pair similarity, which measures consistency across 1,225 pairs of runs (where each run consists of 1,000 documents), shows minimal variation with a standard deviation of 0.00 for both text types. The minimum similarity scores remain high at 0.97 for MD&As and 0.97 for call presentations, indicating that even the least similar run pairs maintain strong semantic alignment.

At the document level, which examines the average similarity of the same document summarized across 1,225 run pairs, we observe slightly more variation with standard deviations of 0.01 for MD&As and 0.02 for call presentations. The minimum similarity scores at this level are 0.90 for

MD&As and 0.88 for call presentations, which, while lower than the run-pair minimums, still represent strong semantic consistency.

Figure B2 illustrates the distributions of these similarity scores. Panel A shows that for MD&As, both run-level and document-level similarities are tightly clustered around the mean of 0.98, with most values falling between 0.96 and 0.99. Panel B demonstrates a similar pattern for call presentations, though with slightly more variance in the document-level similarities. In both cases, the distributions are left-skewed, with the vast majority of similarity scores above 0.95, further confirming the high level of semantic consistency in the generated summaries.

These results suggest that GPT-4o-mini produces highly consistent summaries in terms of semantic content, regardless of whether it is summarizing MD&A sections from 10-Ks or presentation sections from earnings call transcripts. This consistency is particularly notable given the complex and specialized nature of financial texts, indicating that the model maintains reliable performance when handling domain-specific summarization tasks.

B4.2 Length of Summaries

We also assess consistency in terms of the length of the summaries, as measured in word count, because prior studies have used the length of summaries to assess the information content of the original disclosure text, with relatively shorter summaries found to indicate a practice of bloated disclosure (A. Kim et al. 2024a). As shown in Table B5, the consistency in word count demonstrates notable patterns across both document types.

For MD&As, the run-level metrics reveal moderate consistency, with an ICC2 of 0.66 and mean Spearman correlation of 0.64, indicating reasonable rank-order stability across runs. However, the concordance correlation is lower at 0.43, suggesting some systematic differences in absolute word counts between runs. The run-pair Mean Absolute Relative Differences (MARD) of 8.51% quantifies this variation, showing that on average, word counts differ by approximately 8.51% between runs.

For earnings call presentations, we observe generally higher consistency, with an ICC2 of 0.75 and mean Spearman correlation of 0.76, alongside a run-pair MARD of 8.37%. At the document level, the data shows that virtually no MD&A documents (0.00%) receive identical word counts across runs, while a small portion (0.10%) of earnings call presentations maintain identical outputs in length.

Document-wise MARD values are similar to their run-level counterparts, but with notably higher standard deviations (3.38% for MD&As and 4.34% for earnings calls), indicating that consistency varies substantially across individual documents. These findings suggest that while the models maintain reasonable overall consistency in summary length, particularly for earnings calls, there remains meaningful run-to-run variation that researchers should consider when using LLM-generated summaries for disclosure analysis.

Overall, these findings suggest that while the rank ordering of document lengths remains relatively stable across runs (as indicated by the high rank correlations), the exact word counts vary moderately. This variation in length, despite consistent ordering, indicates that the models maintain similar judgments about relative information density across documents but exhibit flexibility in the precise verbosity of individual summaries.

To complement our tabular analysis, we plot the distributions of MARDs for both document types in Figure B3. The distributions reveal important patterns not captured by aggregate statistics alone. While run-level MARDs (left panels) display relatively symmetric and tightly clustered distributions around their means, document-level MARDs (right panels) exhibit notably different characteristics. Though sharing identical means with their run-level counterparts, document-level distributions show substantially greater dispersion and right-skewness. This difference is expected, as run-level metrics represent averages across many documents, naturally reducing variability through aggregation. The document-level distributions highlight that individual documents can experience dramatically different consistency outcomes, with some having MARDs exceeding 20%, particularly for earnings call presentations.

B4.3 Tone

We also assess how the tone of the summaries may vary with runs. To avoid introducing more randomness in our assessment, we measure tone using two deterministic approaches, i.e., L&M word lists (Loughran and McDonald 2011) and FinBERT (Huang et al. 2023).

Table B6 reports the consistency of tone in summaries generated by GPT-4o-mini across multiple runs. For MD&As (Panel A), moderate consistency is observed with both assessment methods, though with notable differences between them. FinBERT shows higher inter-rater agreement with Fleiss' Kappa and Krippendorff's Alpha values of 0.68, compared to 0.55 for L&M. This moderate agreement suggests some variability in tone across runs, despite using the same model. The mean run-pair agreement reaches 84.40% using FinBERT and 73.15% using L&M, indicating that while most run pairs produce similar tone classifications, substantial variation exists.

For earnings call presentations (Panel B), tone consistency improves significantly, with mean run-pair agreement reaching 95.55% (FinBERT) and 91.21% (L&M). Perfect agreement percentages also rise substantially in this context (82.20% for FinBERT vs. 70.30% for L&M), suggesting that the model produces more stable tone classifications for presentation summaries than for MD&As.

The class distribution reveals that GPT-4o-mini-generated summaries are predominantly classified as positive across both document types, especially for presentations (93.90% positive when assessed with FinBERT). This significant sentiment imbalance likely contributes to the high consistency observed, as the model demonstrates a systematic tendency toward positive framing. This explains why presentations show more stable tone classifications than MD&As—likely due to both the model's more pronounced positive framing of presentation content and the more standardized language patterns compared to complex MD&As. While GPT-4o-mini demonstrates reasonable tone consistency across runs, both content type and the model's inherent positive framing tendencies significantly influence consistency metrics—an important consideration when interpreting results in financial text analysis applications.

B5 Text Generation

B5.1 Semantic Similarity

For this task, we ask the model to assume the role of a company executive answering questions posed by financial analysts during an earnings conference call. Compared with summarization, this task presents a greater challenge for maintaining consistency, as it offers more flexibility in response generation, even when provided with background information such as presentation section summaries and previous Q&A exchanges. Similar to our approach with the summarization task, we first assess semantic consistency using cosine similarity based on embeddings generated by the *jina-embeddings-v3* model.

As shown in Table B7, both models demonstrate strong consistency in the text generation task, though with notable differences between them. GPT-4o-mini achieves higher mean similarity scores (0.97) compared to GPT-3.5-turbo (0.94) across both run-pair and document-level metrics. This difference of 0.03 points is statistically significant. While both models maintain high run-pair similarity with minimal standard deviation (0.00), GPT-4o-mini demonstrates considerably better minimum scores (0.96 vs. 0.93), suggesting more reliable consistency across different runs. At the document level, GPT-4o-mini not only maintains a higher mean similarity (0.97 vs. 0.94) but also shows half the standard deviation (0.02 vs. 0.04) of GPT-3.5-turbo, indicating more stable performance across individual documents.

Compared to the summarization task, the mean similarity scores are smaller, consistent with the fact that it is harder to maintain consistency with question answering than summarization. This is particularly evident for GPT-3.5-turbo, where the minimum document-level similarity drops to 0.46, suggesting significant inconsistency in some responses. In contrast, GPT-4o-mini maintains a respectable minimum similarity of 0.76, demonstrating greater reliability even in this more challenging generation context. The interquartile range (P75-P25) is also narrower for GPT-4o-mini (0.02) compared to GPT-3.5-turbo (0.04), further supporting its superior consistency.

The distributional differences between the two models become visually apparent when we plot their performances. As shown in Figure B4, the left panel clearly illustrates that GPT-4o-mini's run-level similarity distribution is tightly clustered around 0.97, while GPT-3.5-turbo shows a broader distribution centered around 0.94-0.95. This separation between the distributions highlights the substantial improvement in consistency achieved by GPT-4o-mini. The right panel's document-level comparison reveals that nearly all data points fall above the 45-degree line, confirming that GPT-4o-mini consistently produces higher similarity scores than GPT-3.5-turbo across the vast majority of documents, with most points clustering in the 0.95-1.00 range for GPT-4o-mini compared to the more dispersed 0.85-0.98 range for GPT-3.5-turbo.

B5.2 Length

We similarly assess the consistency in the length of the answers generated by two models. As shown in Panel B of Table B8, GPT-4o-mini and GPT-3.5-turbo demonstrate different patterns of consistency in answer length across multiple runs.

GPT-4o-mini exhibits superior consistency with an ICC2 of 0.89 and mean Spearman correlation of 0.89, indicating high reliability across runs in maintaining relative answer lengths. The concordance correlation is also strong at 0.79, suggesting minimal systemic differences in absolute word counts between runs. Most notably, GPT-4o-mini shows substantially lower variation in length with a run-pair MARD of only 8.18% compared to GPT-3.5-turbo's 14.91%, representing a difference of 6.73 percentage points.

In contrast, while GPT-3.5-turbo maintains respectable consistency with an ICC2 of 0.76 and mean Spearman correlation of 0.78, its concordance correlation is considerably lower at 0.58. At the document level, both models show similar proportions of documents with identical outputs across runs (0.05% for GPT-4o-mini vs. 0.10% for GPT-3.5-turbo), but GPT-3.5-turbo displays much higher variability in document-wise MARD with a standard deviation of 8.50% compared to GPT-4o-mini's 3.52% – a substantial difference of 4.99 percentage points.

These findings suggest that while both models maintain good consistency in how they answer questions (as evidenced by strong Spearman correlations), GPT-4o-mini provides considerably more stability in the exact word counts of responses. The lower MARD values across all metrics for GPT-4o-mini indicate that it produces more consistent output lengths between runs.

To visualize these differences in length consistency, Figure B5 presents the distribution of MARD values for both models across runs and documents. The left panel shows the run-level MARD distributions with a clear separation between the models, while the right panel displays a document-by-document comparison of MARD values.

As evident from Figure B5, the histograms reveal non-overlapping distributions, with GPT-4o-mini's errors tightly clustered around 8% and GPT-3.5-turbo's centered near 15%. The scatter plot further confirms this pattern at the document level, where points predominantly fall below the diagonal line, indicating that GPT-4o-mini consistently produces lower variation in response length for the same prompts across multiple runs. The dense concentration of points in the lower left quadrant highlights that even when GPT-3.5-turbo performs relatively well (15-25% MARD), GPT-4o-mini still maintains superior consistency (5-15% MARD). The visualizations confirm that GPT-4o-mini provides considerably more stability in the exact word counts of responses.

B5.3 Tone

Similar to the earlier summarization task, we assess the consistency of the tone of the answers generated by two GPT models, and the tone is measured by FinBERT and L&M. Table B9 presents the consistency results. When the tone is measured by FinBERT (Panel A), both models demonstrate moderate consistency, with GPT-4o-mini slightly outperforming GPT-3.5-turbo on most metrics. The inter-rater agreement scores are comparable (0.66 vs. 0.65), but more substantial differences emerge at the document level. GPT-4o-mini achieves notably higher perfect agreement rates (67.00% vs. 49.55%) and mean run-pair agreement (90.64% vs. 85.34%), indicating greater stability in maintaining tone across runs.

Using the L&M Dictionary (Panel B), overall consistency levels are lower for both models compared to FinBERT measurements, with Fleiss' Kappa values of only 0.57 and 0.56 for GPT-4o-mini and GPT-3.5-turbo, respectively. However, the relative performance pattern persists, with GPT-4o-mini maintaining higher consistency across most metrics. Perfect agreement percentages drop significantly under this measurement approach (44.10% vs. 33.70%), suggesting greater sensitivity to lexical variations in the L&M method.

Class distribution analysis reveals that both models produce predominantly positive-toned answers, though GPT-4o-mini generates significantly more positive content across both measurement methods (85.70% vs. 73.65% with FinBERT; 78.40% vs. 68.35% with L&M). Answers generated by GPT-3.5-turbo contain a higher proportion of neutral content, particularly when measured by FinBERT (22.60% vs. 9.45%).

Overall, while both models show moderate consistency in tone across multiple generations, GPT-4o-mini demonstrates superior stability, particularly at the document level, regardless of the measurement approach used. This suggests that more advanced models may produce more consistent tonal characteristics when generating financial text responses.

Our findings also suggest FinBERT provides more consistent tone measurements than fixed word lists when evaluating LLM-generated financial text. Higher agreement metrics across both models when using FinBERT indicates better reproducibility, a critical consideration for research involving stochastic language models. This advantage likely stems from FinBERT's ability to capture contextual nuances and semantic variations that LLMs employ, whereas fixed word lists are limited by predefined vocabulary. For LLM-generated content, where expression patterns may vary across runs while maintaining similar intent, context-aware measurement tools like FinBERT offer more reliable tone consistency evaluations.

B5.4 Prediction of Future Earnings

In this section, we assess the consistency of future earnings predicted by three models: GPT-3.5-turbo, GPT-4o-mini, and GPT-4o. For this task, we ask the models to provide both a directional prediction (increase or decrease) and a point estimate for the next fiscal year's earnings. We first examine the results of directional prediction, which is essentially a binary classification task.

Table B10 presents the consistency analysis for the directional prediction results across 1,000 firm-years, each classified 50 times by all three models. All models demonstrate high consistency, with Fleiss' Kappa and Krippendorff's Alpha values of 0.97 for GPT-3.5-turbo, 0.99 for GPT-4o-mini, and 0.98 for GPT-4o, all well above the 0.8 reliability threshold.

The run-level metrics confirm this high consistency, with run-pair agreement ranging from 98.54% (GPT-3.5-turbo) to 99.33% (GPT-4o-mini). At the document level, perfect agreement rates are 93.70% for GPT-3.5-turbo, 97.30% for GPT-4o-mini, and 94.90% for GPT-4o. Mean document-wise agreement and majority class strength follow similar patterns, with GPT-4o-mini consistently showing the highest consistency despite being less advanced than GPT-4o.

For class-specific performance, all models maintain high agreement rates for both increase and decrease predictions. The class distribution varies notably across models: GPT-4o predicts increases 54.20% of the time, GPT-4o-mini 45.10%, and GPT-3.5-turbo 52.80%.

Interestingly, our findings reveal that while all three models demonstrate high consistency, GPT-4o-mini outperforms the more advanced GPT-4o in several consistency metrics. This suggests that model advancement does not necessarily correlate with increased consistency in this specific prediction task, though all models achieve reliability levels that would be considered excellent by conventional standards.

We further assess the consistency in the point estimates of future earnings generated by the three models. As shown in Table B11, the models demonstrate varying degrees of consistency across multiple reliability metrics. GPT-4o exhibits exceptional consistency with almost perfect scores of 1.00 across ICC2, mean and median concordance correlation, and mean and median Spearman correlation. This indicates almost perfect reliability across runs in maintaining both absolute earnings values and relative rankings of earnings estimates. Most notably, GPT-4o shows minimal variation with a mean run-pair MARD of only 2.05%, substantially lower than GPT-4o-mini's 6.18% and GPT-3.5-turbo's 9.89% – representing differences of 4.13 and 7.84 percentage points respectively.

GPT-4o-mini demonstrates great consistency with an ICC2 of 0.86 and mean Spearman correlation of 0.97, indicating strong reliability in maintaining relative earnings rankings across runs. However, its mean concordance correlation is lower at 0.76, suggesting some systemic differences in absolute earnings values between runs. The standard deviation of concordance correlation (0.15) also indicates greater variability compared to GPT-4o's near-perfect 0.01.

GPT-3.5-turbo, while maintaining a respectable ICC2 of 0.85 and strong mean Spearman correlation of 0.97, shows the lowest mean concordance correlation at 0.75 and highest standard deviation of concordance correlation at 0.28. This suggests greater inconsistency in absolute earnings estimates between runs, despite maintaining consistent relative rankings.

At the document (firm-year) level, GPT-4o produces the highest proportion of documents with identical point estimates across runs (29.20%) compared to GPT-4o-mini (32.80%) and GPT-3.5-turbo (10.90%). However, the document-level MARD metrics reveal substantial differences in variability. GPT-3.5-turbo displays the highest mean document-wise MARD at 9.89% and exhibits extreme variability with a standard deviation of 20.69%. In contrast, GPT-4o-mini shows moderate variability (standard deviation of 18.10%), while GPT-4o demonstrates remarkably low variability (standard deviation of 7.75%).

These findings suggest that while all three models maintain strong consistency in how they rank firm-years by earnings estimates (as evidenced by similar high Spearman correlations), GPT-4o provides significantly more stability in the exact earnings values of responses. The progressively higher MARD values from GPT-4o to GPT-4o-mini to GPT-3.5-turbo indicate a clear correlation between model advancement and consistency in this quantitative prediction task, making the more advanced models potentially more reliable for applications where consistent numerical estimates are critical.

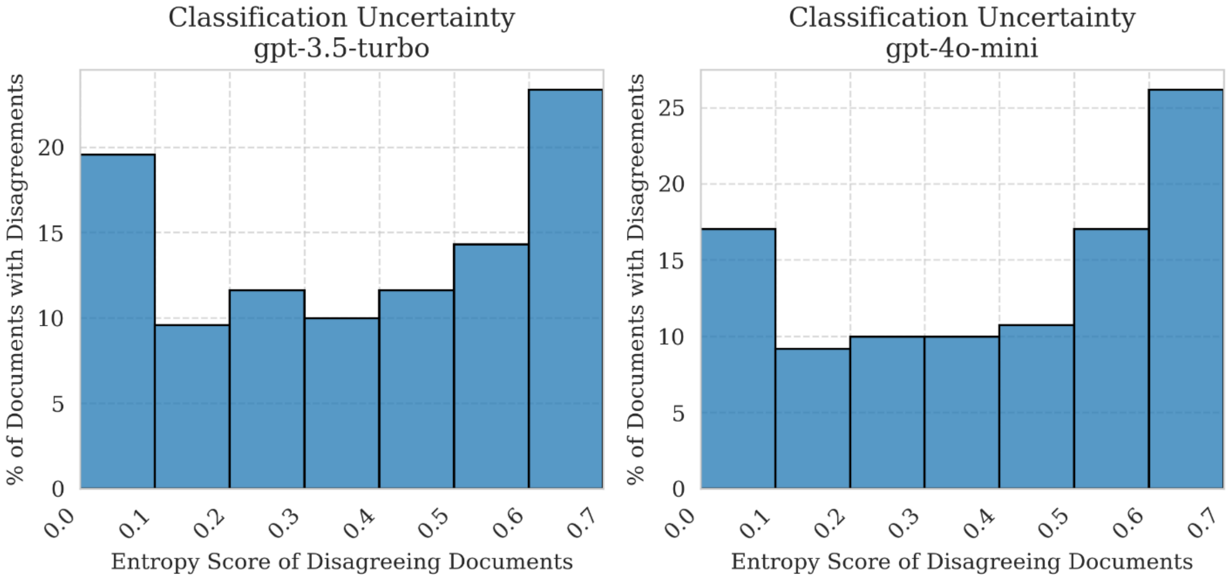
To better understand the difference in performance among these models, we plot the distributions of concordance correlation coefficients (CCC) and MARD values at both run and document (firm-year) levels in Figure B6. Panel A illustrates the distribution of CCC values, revealing that GPT-4o maintains near-perfect consistency with minimal dispersion around 1.0. In contrast, GPT-4o-mini shows a wider, unimodal distribution centered around 0.8, while GPT-3.5-turbo exhibits a distinctive bimodal distribution with peaks at approximately 0.9 and 0.4, indicating two distinct patterns of behavior across different runs.

Panel B presents the run-level pairwise MARD distributions, providing a clear visualization of the consistency gap between models. GPT-4o demonstrates a remarkably narrow distribution centered at approximately 0.02 (2%), with minimal dispersion. GPT-4o-mini shows a broader, unimodal distribution centered around 0.06 (6%), while GPT-3.5-turbo exhibits the widest distribution, centered around 0.09-0.10 (9-10%) with a notable right tail extending beyond 0.15 (15%). The clear separation between these distributions confirms a substantial stepwise improvement in consistency as we move from GPT-3.5-turbo to GPT-4o-mini to GPT-4o.

Panel C provides the Empirical Cumulative Distribution Function (ECDF) of document-level MARD values, offering insight into the probability distribution of consistency across the document corpus. The steeper initial slope of GPT-4o's curve (blue) indicates that a larger proportion of its documents have exceptionally low MARD values compared to the other models. Specifically, approximately 80% of documents processed by GPT-4o have MARD values below 0.10 (10%), compared to approximately 70% for GPT-4o-mini and 60% for GPT-3.5-turbo at the same threshold. This ECDF visualization clearly demonstrates GPT-4o's stochastic dominance in consistency performance across the entire distribution of documents.

Collectively, these visualizations provide compelling evidence that model sophistication correlates strongly with consistency in numerical predictions, with GPT-4o demonstrating superior reliability across all metrics examined. The progressively wider distributions observed for less sophisticated models suggest that consistency should be a critical consideration when selecting models for tasks requiring precise numerical outputs, such as financial analysis and forecasting applications.

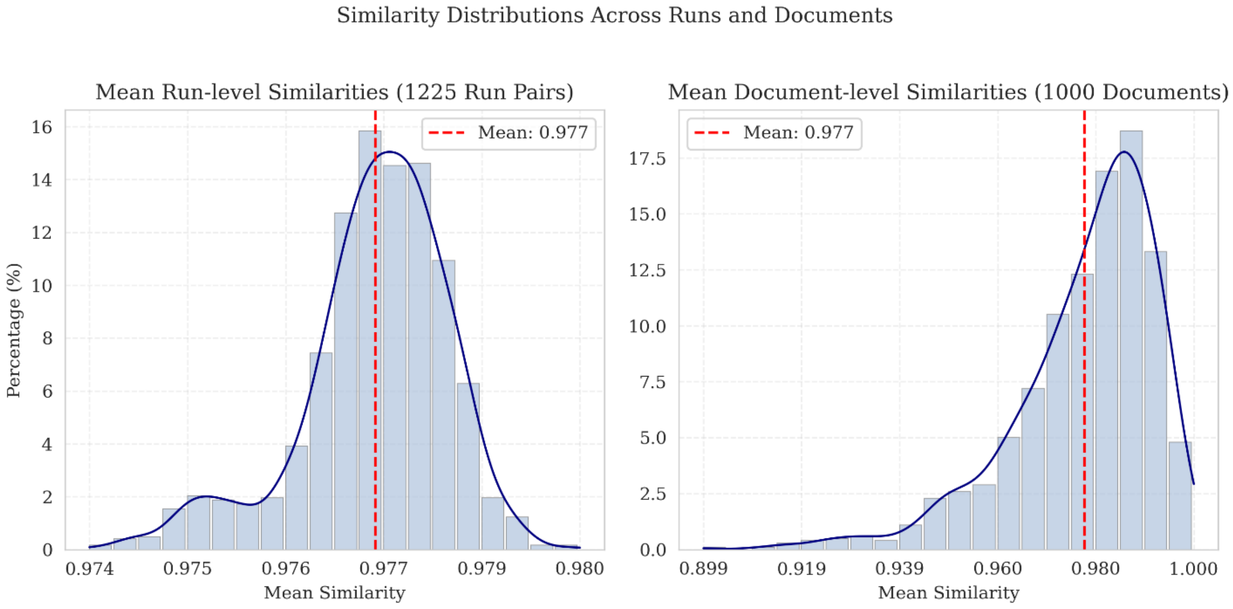
Figure B1 Classification Uncertainty for Documents with Disagreements (Binary Classification)



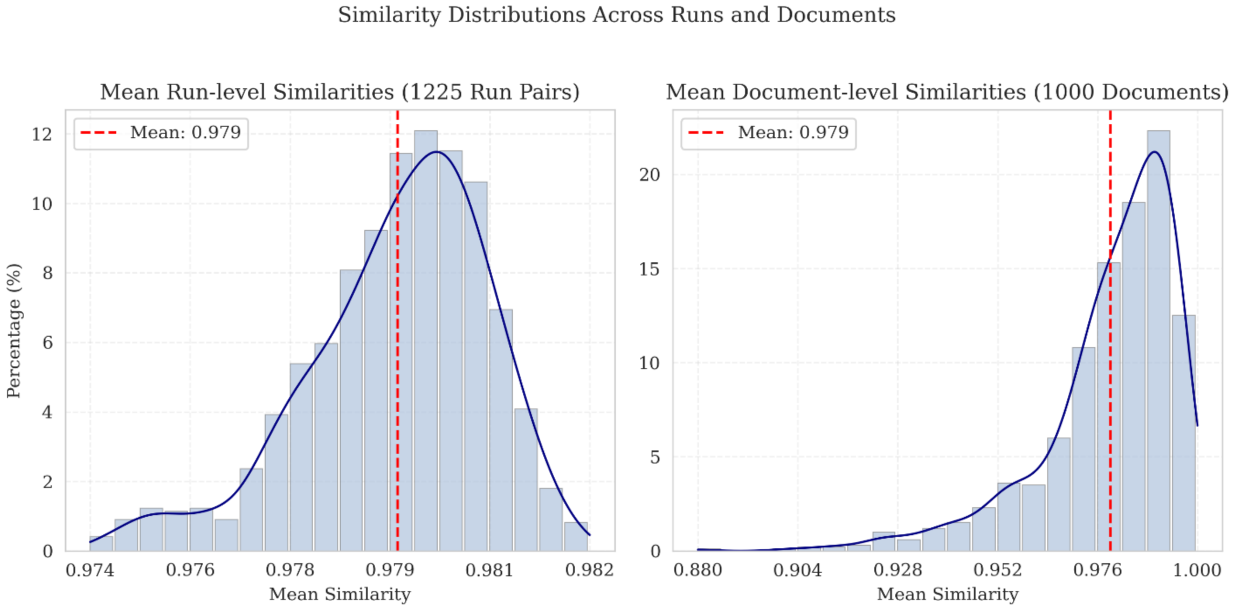
Note: Distribution of entropy scores for documents with at least one disagreement (n=783 for GPT-3.5-turbo, n=382 for GPT-4o-mini).

Figure B2 Distribution of Semantic Similarity Scores of Generated Summaries

Panel A: Similarity Distributions for MD&A Summaries

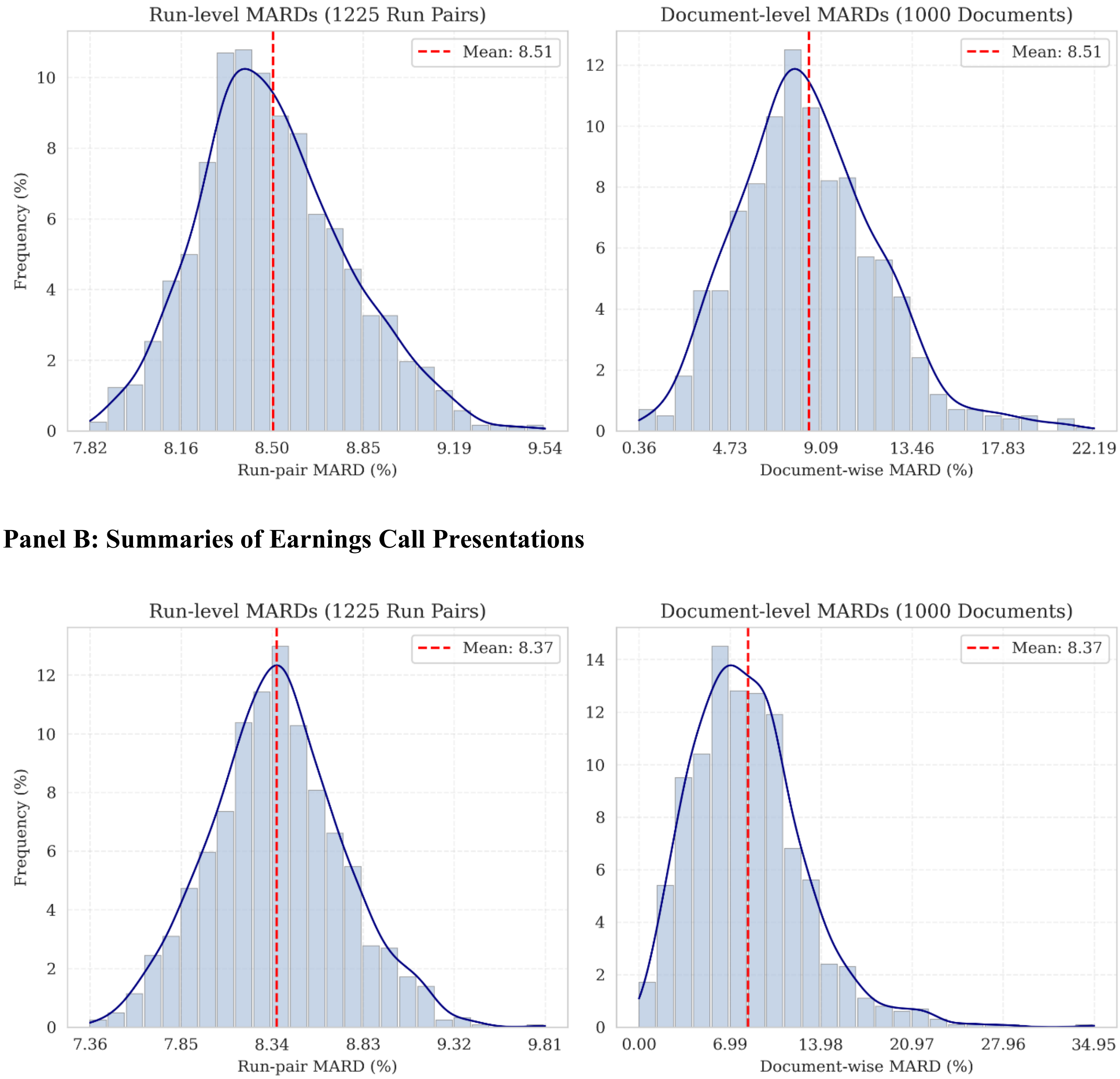


Panel B: Similarity Distributions for Summaries of Earnings Call Presentations



Note: See Appendix B for metric definitions.

Figure B3 Mean Absolute Relative Differences for Lengths of Summaries

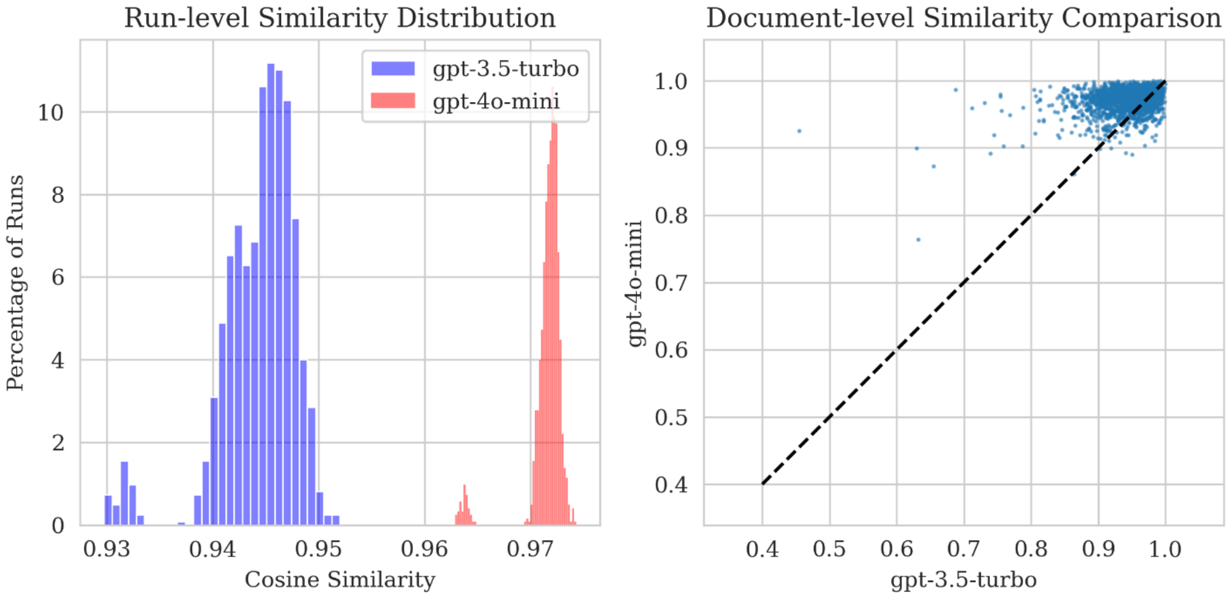


Panel A: Summaries of MD&As

Panel B: Summaries of Earnings Call Presentations

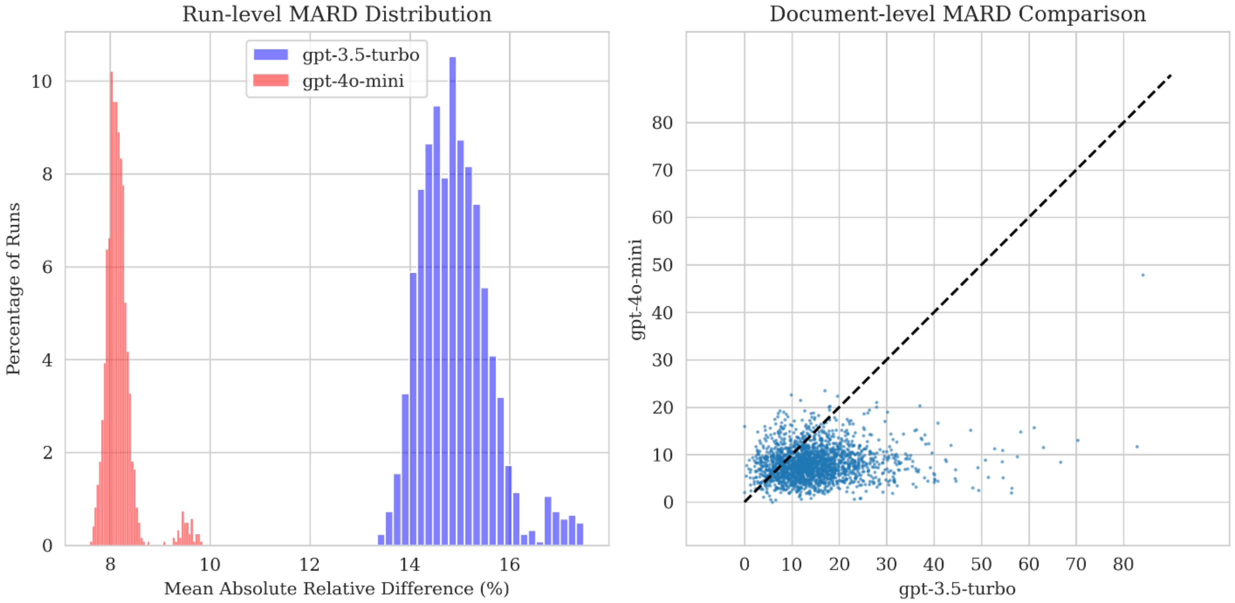
Note: See Appendix B for metric definitions.

Figure B4 Comparative Analysis of Model Consistency in Financial Q&A Generation



Note: The left panel displays the distribution of run-level similarity scores for both models. The right panel presents a document-by-document comparison where each point represents similarity scores for the same document generated by both models. Points above the dashed 45-degree line indicate higher consistency from GPT-4o-mini, which is the case for nearly all documents analyzed. See Appendix B for metric definitions.

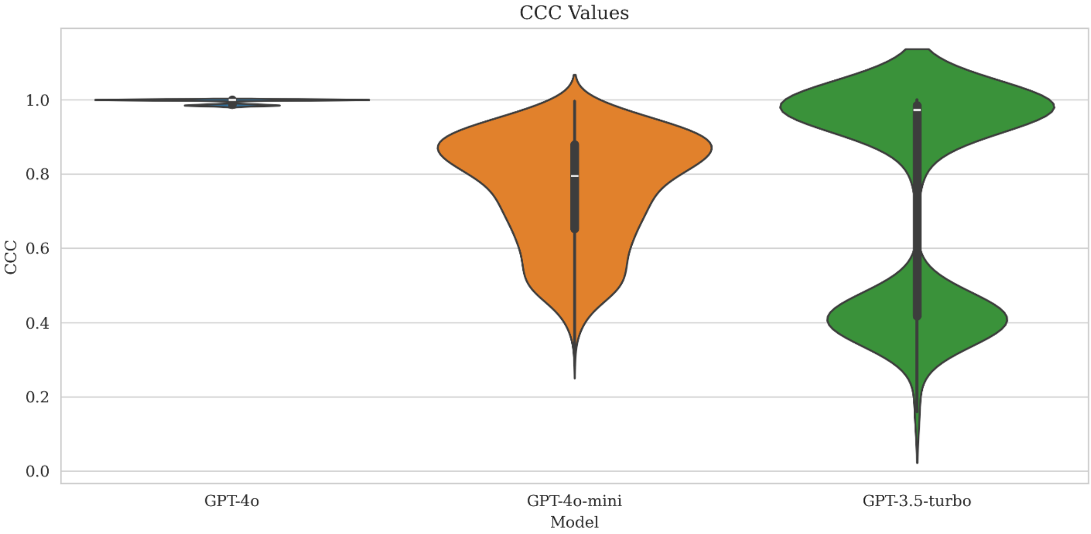
Figure B5 Comparative Analysis of Model Consistency in Length of Question Answering



Note: The left panel shows the distribution of Mean Absolute Relative Difference (MARD) values for response lengths across multiple runs for GPT-3.5-turbo (blue) and GPT-4o-mini (red). The right panel displays document-level MARD comparisons, with each point representing a single document's MARD values for both models. Points below the diagonal indicate lower variation in GPT-4o-mini's response lengths compared to GPT-3.5-turbo for the same documents. See Appendix B for metric definitions.

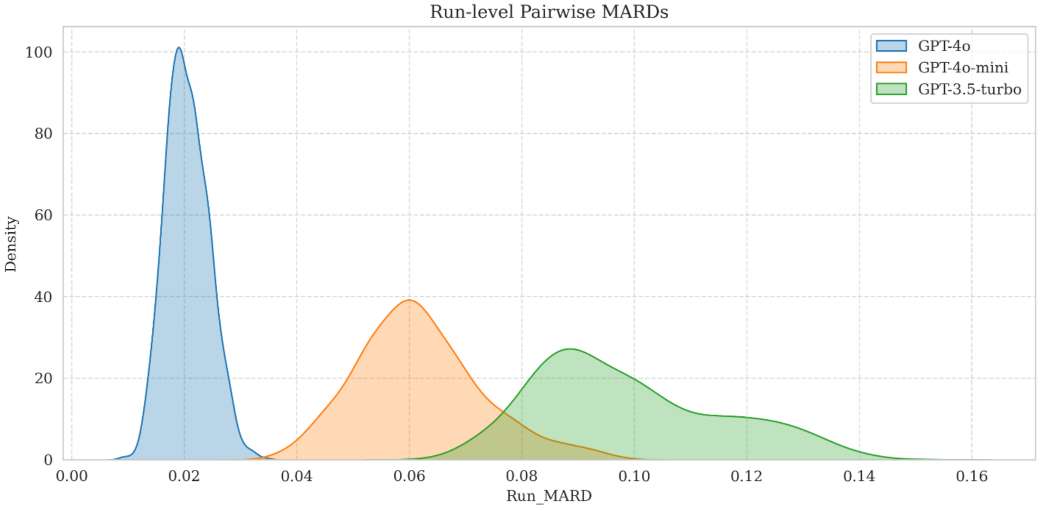
Figure B6 Consistency Analysis of Model Performance Across Multiple Runs for Prediction Task

Panel A: Concordance Correlation Coefficient (CCC) Distributions by Model



Note: Violin plots showing the distribution of pairwise concordance correlation coefficients between runs. GPT-4o exhibits near-perfect consistency (values clustered at 1.0), while GPT-4o-mini shows moderate variability. GPT-3.5-turbo displays a bimodal distribution, indicating two distinct performance patterns. See Appendix B for metric definitions.

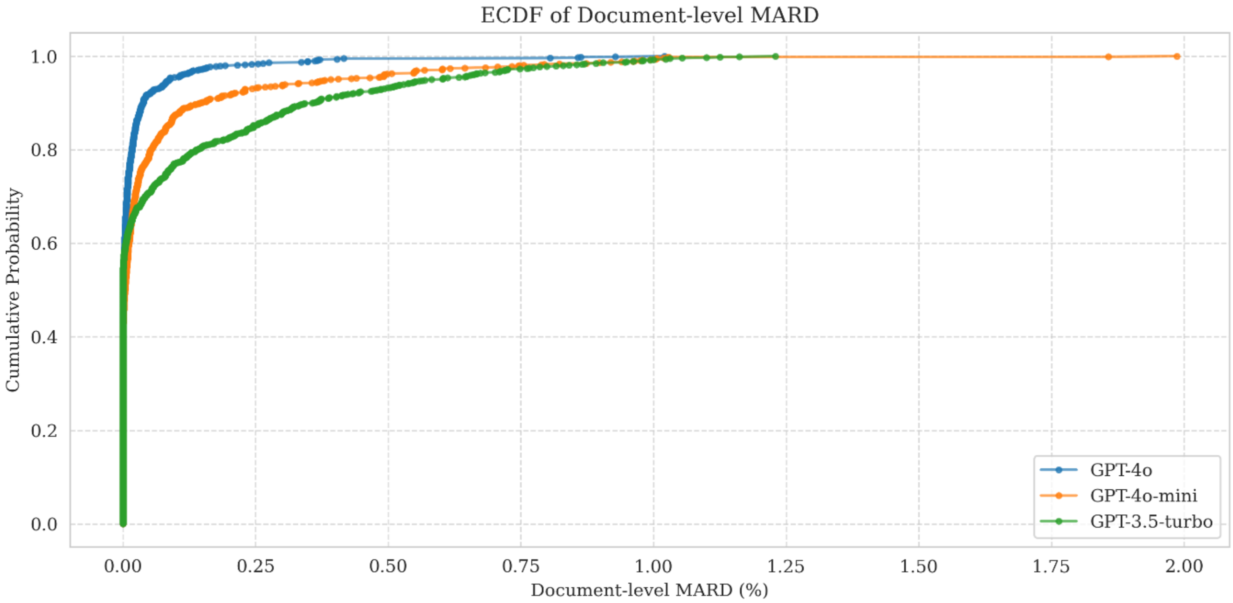
Panel B: Run-level Mean Absolute Relative Difference (MARD) Distributions



Note: Density plots of pairwise MARD between runs for each model. Lower MARD values indicate higher consistency. GPT-4o shows a narrow distribution centered around 2%, GPT-4o-mini around 6%, and GPT-3.5-turbo around 10%, demonstrating a clear progression in consistency with model sophistication. See Appendix B for metric definitions.

Figure B6 Consistency Analysis of Model Performance Across Multiple Runs for Prediction (Continued)

Panel C: Empirical Cumulative Distribution Function (ECDF) of Document-level MARD



Note: ECDF curves showing the cumulative probability of document-level MARD values. The leftward shift of GPT-4o's curve demonstrates stochastic dominance, with approximately 80% of documents having MARD values below 10%, compared to 70% for GPT-4o-mini and 60% for GPT-3.5-turbo. See Appendix B for metric definitions.

Table B1 Consistency Results for Binary Classification of Forward-looking Statements

Panel A: Overall Inter-rater and Document-level Agreements

Metric	GPT-4o-mini	GPT-3.5-turbo	Difference
Sample Size			
Number of documents	10,000	10,000	-
Number of runs	50	50	-
Overall Inter-rater Agreements			
Fleiss' Kappa	0.97	0.93	0.04
Krippendorff's Alpha	0.97	0.93	0.04
Run-level Agreements			
Mean run-pair Cohen's Kappa	0.97	0.93	0.04***
Mean run-pair agreement (%)	99.00	98.07	0.93***
Document-level Agreements			
Percentage of perfect agreement (%)	96.18	92.17	4.01
Mean document-wise agreement (%)	99.00	98.07	0.93***
Mean majority class strength (%)	99.30	98.67	0.64***
Mean classification uncertainty	0.40	0.38	0.02
Class-specific Agreements			
Forward-looking (%)	97.36	94.00	3.36
Non-forward looking (%)	99.10	98.44	0.66
Class Distribution			
Forward-looking (%)	24.30	17.76	6.54
Non-forward looking (%)	75.70	82.24	-6.54

Note: Five key metrics are tested for statistical differences between models (GPT-4o-mini minus GPT-3.5-turbo): Mean run-pair Cohen's Kappa, Mean run-pair agreement, Mean document-wise agreement, Mean majority class strength, and Mean classification uncertainty. Paired tests are used when measurements can be matched across models, while independent tests are used for aggregated metrics. Stars indicate statistical significance: *** p<0.01, ** p<0.05, * p<0.10. See Appendix B for metric definitions.

Table B1 Consistency Results for Binary Classification of Forward-looking Statements (Continued)

Panel B: Detailed Run-level Agreements

Model	Metric	N	Mean	Mdn	Std	Min	P25	P75	Max
GPT-3.5-turbo	Run-pair Cohen's Kappa score	1225	0.93	0.93	0.01	0.90	0.93	0.94	0.95
	Run-pair agreement (%)	1225	98.07	98.09	0.20	97.14	97.98	98.19	98.52
GPT-4o-mini	Run-pair Cohen's Kappa	1225	0.97	0.97	0.00	0.96	0.97	0.97	0.98
	Run-pair agreement (%)	1225	99.00	99.00	0.09	98.66	98.93	99.06	99.25

Note: See Appendix B for metric definitions.

Panel C: Detailed Document-level Agreements

Model	Metric	N	Mean	Mdn	Std	Min	P25	P75	Max
GPT-3.5-turbo	Document-wise agreement (%)	10,000	98.07	100	8.11	48.98	100	100	100
	Majority class strength (%)	10,000	98.67	100	6.08	50.00	100	100	100
	Classification uncertainty	783	0.38	0.37	0.21	0.10	0.17	0.58	0.69
GPT-4o-mini	Document-wise agreement (%)	10,000	99.00	100	6.00	48.98	100	100	100
	Majority class strength (%)	10,000	99.30	100	4.47	50.00	100	100	100
	Classification uncertainty	382	0.40	0.40	0.21	0.10	0.17	0.61	0.69

Note: See Appendix B for metric definitions.

Table B2 Consistency Results for Multi-class Classification of FOMC Statements

Panel A: Overall Inter-rater and Document-level Agreements

Metric	GPT-4o-mini	GPT-3.5-turbo	Difference
Sample Size			
Number of documents	1,096	1,096	-
Number of runs	50	50	-
Overall Inter-rater Agreements			
Fleiss' Kappa	0.86	0.91	-0.05
Krippendorff's Alpha	0.86	0.91	-0.05
Run-level Agreements			
Mean run-pair Cohen's Kappa	0.86	0.91	-0.05***
Mean run-pair agreement (%)	88.93	92.96	-4.03***
Document-level Agreements			
Percentage of perfect agreement (%)	55.66	65.33	-9.67
Mean document-wise agreement (%)	88.93	92.96	-4.03***
Mean majority class strength (%)	92.26	95.32	-3.06***
Mean classification uncertainty	0.40	0.34	0.06***
Class-specific Agreements			
Hawkish (%)	90.29	93.13	-2.84
Mostly Hawkish (%)	87.19	92.74	-5.55
Neutral (%)	91.83	93.66	-1.83
Mostly Dovish (%)	84.90	91.34	-6.44
Dovish (%)	88.61	90.22	-1.61
Class Distribution			
Hawkish (%)	8.30	6.02	2.28
Mostly Hawkish (%)	18.70	23.36	-4.65
Neutral (%)	32.57	34.31	-1.73
Mostly Dovish (%)	19.07	25.36	-6.30
Dovish (%)	21.35	10.95	10.40

Note: See the note from Table B1.

Table B2 Consistency Results for Multi-class Classification of FOMC Statements (Continued)

Panel B: Detailed Run-level Agreements

Model	Metric	N	Mean	Mdn	Std	Min	P25	P75	Max
GPT-3.5-turbo	Run-pair Cohen's Kappa	1,225	0.91	0.92	0.06	0.69	0.91	0.93	0.96
	Run-pair agreement (%)	1,225	92.96	94.07	4.28	77.28	93.52	94.62	96.81
GPT-4o-mini	Run-pair Cohen's Kappa	1,225	0.86	0.87	0.06	0.63	0.86	0.88	0.92
	Run-pair agreement (%)	1,225	88.93	90.15	4.51	71.81	89.32	90.78	93.80

Note: See Appendix B for metric definitions.

Panel C: Detailed Document-level Agreements

Model	Metric	N	Mean	Mdn	Std	Min	P25	P75	Max
GPT-3.5-turbo	Document-wise agreement (%)	1,096	92.96	100	13.80	36.24	92.16	100	100
	Majority class strength (%)	1,096	95.32	100	10.34	42.00	96.00	100	100
	Classification uncertainty	380	0.34	0.23	0.23	0.10	0.17	0.53	1.04
GPT-4o-mini	Document-wise agreement (%)	1,096	88.93	100	17.20	33.31	84.98	100	100
	Majority class strength (%)	1,096	92.26	100	13.49	42.00	92.00	100	100
	Classification uncertainty	486	0.40	0.37	0.24	0.10	0.17	0.63	1.14

Note: See Appendix B for metric definitions.

Table B3 Consistency Results for Sentiment Analysis

Panel A: Overall Inter-rater and Document-level Agreements

	News		MD&A		Call Presentation		Call Q&A	
Metric	**4oM**	**3.5T**	**4oM**	**3.5T**	**4oM**	**3.5T**	**4oM**	**3.5T**
Sample Size								
Number of documents	4,838	4,838	10,000	10,000	2,000	2,000	2,000	2,000
Number of runs	50	50	50	50	50	50	50	50
Overall Inter-rater Agreements								
Fleiss' Kappa	0.96	0.94	0.97	0.95	0.98	0.96	0.97	0.95
Krippendorff's Alpha	0.96	0.94	0.97	0.95	0.98	0.96	0.97	0.95
Run-level Agreements								
Mean run-pair Cohen's Kappa	0.96	0.94	0.97	0.95	0.98	0.96	0.97	0.95
Mean run-pair agreement (%)	97.75	96.38	97.97	96.92	98.71	97.47	98.55	97.49
Document-level Agreements								
Percentage of perfect agreement (%)	88.28	81.09	92.46	88.53	95.05	90.65	93.80	90.10
Mean document-wise agreement (%)	97.75	96.38	97.97	96.92	98.71	97.47	98.55	97.49
Mean majority class strength (%)	98.48	97.60	98.57	97.82	99.09	98.23	99.01	98.24
Mean classification uncertainty	0.31	0.31	0.41	0.41	0.40	0.42	0.37	0.39
Class-specific Agreements								
Positive (%)	97.42	95.94	97.30	95.27	99.15	97.74	98.80	97.24
Neutral (%)	96.74	95.44	97.59	96.62	96.52	95.21	97.39	96.97
Negative (%)	98.32	97.39	96.61	95.28	96.97	95.99	96.64	92.69
Class Distribution								
Positive (%)	44.19	41.22	27.64	21.44	64.15	56.65	55.20	40.50
Neutral (%)	39.85	42.08	51.38	55.93	23.10	32.15	34.65	53.25
Negative (%)	15.96	16.70	20.98	22.63	12.75	11.20	10.15	6.25

Note: 'News' refers to sentences from news articles, 'MD&A' to sentences from MD&A sections, 'Pre' to paragraphs from earnings call presentations, and 'Q&A' to paragraphs from Q&A sections of earnings calls. '4oM' represents GPT-4o-mini, and '3.5T' represents GPT-3.5-turbo. For mean variables, all differences between models for the same text type are statistically significant ($p < 0.01$) except for mean classification uncertainty. See Appendix B for metric definitions.

Table B3 Consistency Results for Sentiment Analysis (Continued)

Panel B: Detailed Run-level Agreements

Model	Metric	N	Mean	Mdn	Std	Min	P25	P75	Max
News - 3.5T	Run-pair Cohen's Kappa	1,225	0.94	0.95	0.04	0.81	0.95	0.96	0.96
	Run-pair agreement (%)	1,225	96.38	97.04	2.32	88.07	96.86	97.19	97.81
News - 4oM	Run-pair Cohen's Kappa	1,225	0.96	0.97	0.02	0.88	0.97	0.97	0.98
	Run-pair agreement (%)	1,225	97.75	98.18	1.51	92.27	98.04	98.28	98.68
MD&A- 3.5T	Run-pair Cohen's Kappa	1,225	0.95	0.95	0.00	0.94	0.95	0.95	0.96
	Run-pair agreement (%)	1,225	96.92	96.92	0.18	96.20	96.81	97.03	97.43
MD&A - 4oM	Run-pair Cohen's Kappa	1,225	0.97	0.97	0.00	0.96	0.97	0.97	0.98
	Run-pair agreement (%)	1,225	97.97	97.97	0.11	97.57	97.89	98.05	98.47
Pre- 3.5T	Run-pair Cohen's Kappa	1,225	0.96	0.96	0.01	0.93	0.95	0.96	0.98
	Run-pair agreement (%)	1,225	97.47	97.50	0.46	96.10	97.15	97.80	98.65
Pre- 4oM	Run-pair Cohen's Kappa	1,225	0.98	0.97	0.01	0.96	0.97	0.98	0.99
	Run-pair agreement (%)	1,225	98.71	98.70	0.26	97.80	98.55	98.90	99.40
Q&A- 3.5T	Run-pair Cohen's Kappa	1,225	0.95	0.96	0.01	0.92	0.95	0.96	0.98
	Run-pair agreement (%)	1,225	97.49	97.60	0.55	95.40	97.25	97.85	98.65
Q&A - 4oM	Run-pair Cohen's Kappa	1,225	0.97	0.98	0.01	0.94	0.97	0.98	0.99
	Run-pair agreement (%)	1,225	98.55	98.60	0.38	96.75	98.40	98.80	99.60

Note: 'News' refers to sentences from news articles, 'MD&A' to sentences from MD&A sections, 'Pre' to paragraphs from earnings call presentations, and 'Q&A' to paragraphs from Q&A sections of earnings calls. '4oM' represents GPT-4o-mini, and '3.5T' represents GPT-3.5-turbo. N=1,225 represents all possible run pairs ((50 × 49)/2). P25 and P75 represent the 25th and 75th percentiles, respectively. See Appendix B for metric definitions.

Table B3 Consistency Results for Sentiment Analysis (Continued)

Panel C: Detailed Document-level Agreements

Model	Metric	N	Mean	Mdn	Std	Min	P25	P75	Max
News-3.5T	Document-wise agreement (%)	4,838	96.38	100	10.12	45.88	100	100	100
	Majority class strength (%)	4,838	97.60	100	7.53	50.00	100	100	100
	Classification uncertainty	915	0.31	0.23	0.20	0.10	0.17	0.47	0.83
News-4oM	Document-wise agreement (%)	4,838	97.75	100	8.31	48.98	100	100	100
	Majority class strength (%)	4,838	98.48	100	6.23	50.00	100	100	100
	Classification uncertainty	567	0.31	0.17	0.21	0.10	0.17	0.53	0.69
MD&A-3.5T	Document-wise agreement (%)	10,000	96.92	100	10.32	32.33	100	100	100
	Majority class strength (%)	10,000	97.82	100	7.92	38.00	100	100	100
	Classification uncertainty	1,147	0.41	0.44	0.22	0.10	0.23	0.61	1.09
MD&A-4oM	Document-wise agreement (%)	10,000	97.97	100	8.49	32.00	100	100	100
	Majority class strength (%)	10,000	98.57	100	6.47	34.00	100	100	100
	Classification uncertainty	754	0.41	0.44	0.21	0.10	0.23	0.61	1.10
Pre-3.5T	Document-wise agreement (%)	2,000	97.47	100	9.45	43.10	100	100	100
	Majority class strength (%)	2,000	98.23	100	7.10	50.00	100	100	100
	Classification uncertainty	187	0.42	0.44	0.22	0.10	0.17	0.62	0.90
Pre-4oM	Document-wise agreement (%)	2,000	98.71	100	6.86	33.14	100	100	100
	Majority class strength (%)	2,000	99.09	100	5.27	42.00	100	100	100
	Classification uncertainty	99	0.40	0.40	0.23	0.10	0.17	0.62	1.08
Q&A-3.5T	Document-wise agreement (%)	2,000	97.49	100	9.29	48.98	100	100	100
	Majority class strength (%)	2,000	98.24	100	7.07	50.00	100	100	100
	Classification uncertainty	198	0.39	0.40	0.22	0.10	0.17	0.61	0.69
Q&A-4oM	Document-wise agreement (%)	2,000	98.55	100	7.09	32.49	100	100	100
	Majority class strength (%)	2,000	99.01	100	5.31	38.00	100	100	100
	Classification uncertainty	124	0.37	0.33	0.23	0.10	0.17	0.59	1.09

Note: 'News' refers to sentences from news articles, 'MD&A' to sentences from MD&A sections, 'Pre' to paragraphs from earnings call presentations, and 'Q&A' to paragraphs from Q&A sections of earnings calls. '4oM' represents GPT-4o-mini, and '3.5T' represents GPT-3.5-turbo. Classification uncertainty is calculated only for documents with disagreements. P25 and P75 represent the 25th and 75th percentiles, respectively. See Appendix B for metric definitions.

Table B4 Semantic Similarity of Generated Summaries

Text	Metric	N	Mean	Mdn	Std	Min	P25	P75	Max
MD&A	Run-pair Similarity	1,225	0.98	0.98	0.00	0.97	0.98	0.98	0.98
MD&A	Document-level Similarity	1,000	0.98	0.98	0.01	0.90	0.97	0.99	1.00
Call presentations	Run-pair Similarity	1,225	0.98	0.98	0.00	0.97	0.98	0.98	0.98
Call presentations	Document-level Similarity	1,000	0.98	0.98	0.02	0.88	0.97	0.99	1.00

Note: "Run-pair similarity" represents the similarity score between output from different model runs. Since each run consists of 1,000 documents, it is calculated as the average similarity across these 1,000 documents. "Document-level similarity" measures the average similarity between summaries of the same document across 1,225 run pairs. All similarity scores are calculated using cosine similarity of jina-embeddings-v3 embeddings. Higher values indicate greater semantic consistency between summaries. See Appendix B for metric definitions.

Table B5 Consistency Results for Lengths of Summaries of MD&As and Earnings Call Presentations

Metric	MD&As (1)	Earnings Call Presentations (2)
Run-level metrics		
ICC2	0.66	0.75
Mean concordance correlation	0.43	0.56
Median concordance correlation	0.43	0.56
Std dev of concordance correlation	0.03	0.03
Mean Pearson correlation	0.66	0.75
Median Pearson correlation	0.66	0.75
Std dev of Pearson correlation	0.02	0.02
Mean Spearman correlation	0.64	0.76
Median Spearman correlation	0.64	0.76
Std dev of Spearman correlation	0.02	0.02
Run-pair MARD (%)	8.51	8.37
Median run-pair MARD (%)	8.48	8.37
Std dev of run-pair MARD (%)	0.28	0.35
Document-level metrics		
Documents with length (%)	0.00	0.10
Document-wise MARD (%)	8.51	8.37
Median document-wise MARD (%)	8.20	7.85
Std dev of document-wise MARD (%)	3.38	4.34

Note: This table presents metrics quantifying the consistency of model outputs across multiple random seed runs. Higher values for ICC2, concordance correlation, and Spearman correlation indicate better consistency between runs. Lower values for MARD (Mean Absolute Relative Difference) metrics indicate less variation between runs. The analysis is conducted at two levels: (1) run-level metrics assess overall consistency between different executions of the model, and (2) document-level metrics evaluate consistency of predictions for individual documents across runs. Values are reported separately for MD&As (Management Discussion & Analysis) and earnings call presentations to facilitate comparison across different text types. See Appendix B for metric definitions.

Table B6 Consistency Results for Tone of Summaries

Panel A: Summaries of MD&As

Metric	FinBERT	L&M	Difference
Sample Size			
Number of documents	1,000	1,000	
Number of runs	50	50	
Overall Inter-rater Agreements			
Fleiss' Kappa	0.68	0.55	-0.13
Krippendorff's Alpha	0.68	0.55	-0.13
Run-level Agreements			
Mean run-pair Cohen's Kappa	0.68	0.55	-0.13***
Mean run-pair agreement (%)	84.40	73.15	-11.25***
Document-level Agreements			
Percentage of perfect agreement (%)	48.10	28.40	-19.70
Mean document-wise agreement (%)	84.40	73.15	-11.25***
Mean majority class strength (%)	89.03	80.67	-8.35***
Mean classification uncertainty	0.48	0.62	0.14***
Class-specific Agreements			
Positive (%)	88.39	77.93	-10.45
Neutral (%)	76.15	72.30	-3.85
Negative (%)	81.19	74.13	-7.05
Class Distribution			
Positive (%)	67.50	54.80	-12.70
Neutral (%)	6.50	9.10	2.60
Negative (%)	26.00	36.10	10.10

Note: FinBERT indicates the tone is measured by using FinBERT, whereas L&M indicates that the tone is measured by using the Loughran & McDonald sentiment dictionary. See Table B1 for additional notes, and see Appendix B for metric definitions.

Table B6 Consistency Results for Tone of Summaries (Continued)

Panel B: Summaries of Earnings Call Presentations

Metric	FinBERT	L&M	Difference
Sample Size			
Number of documents	1,000	1,000	
Number of runs	50	50	
Overall Inter-rater Agreements			
Fleiss' Kappa	0.65	0.59	-0.07
Krippendorff's Alpha	0.65	0.59	-0.07
Run-level Agreements			
Mean run-pair Cohen's Kappa	0.65	0.59	-0.07***
Mean run-pair agreement (%)	95.55	91.21	-4.34***
Document-level Agreements			
Percentage of perfect agreement (%)	82.20	70.30	-11.90
Mean document-wise agreement (%)	95.55	91.21	-4.34***
Mean majority class strength (%)	96.92	93.88	-3.04***
Mean classification uncertainty	0.39	0.50	0.10***
Class-specific Agreements			
Positive (%)	96.88	93.86	-3.02
Neutral (%)	84.67	79.03	-5.64
Negative (%)	82.66	73.11	-9.55
Class Distribution			
Positive (%)	93.90	90.70	-3.20
Neutral (%)	1.70	4.10	2.40
Negative (%)	4.40	5.20	0.80

Note: FinBERT indicates the tone is measured by using FinBERT, whereas L&M indicates that the tone is measured by using the Loughran & McDonald sentiment dictionary. See Table B1 for additional notes, and see Appendix B for metric definitions.

Table B7 Semantic Similarity of Q&A Answers

Model	Metric	N	Mean	Mdn	Std	Min	P25	P75	Max
GPT-3.5-turbo	Run-pair Similarity	1,225	0.94	0.95	0.00	0.93	0.94	0.95	0.95
GPT-3.5-turbo	Document-level Similarity	2,000	0.94	0.95	0.04	0.46	0.93	0.97	1.00
GPT-4o-mini	Run-pair Similarity	1,225	0.97	0.97	0.00	0.96	0.97	0.97	0.97
GPT-4o-mini	Document-level Similarity	1,000	0.97	0.97	0.02	0.76	0.96	0.98	1.00

Note: The similarity measures are based on cosine similarity and are calculated in the same way as Table B4. The difference between the two models is statistically significant for both run-pair similarity and document-level similarity. The former is tested using an independent t-test and the latter using a paired t-test. See Appendix B for metric definitions.

Table B8 Consistency Results for Lengths of Q&A Answers

Metric	GPT-4o-mini (1)	GPT-3.5-turbo (2)	Difference (3)
Run-level metrics			
ICC2	0.89	0.76	0.13
Mean concordance correlation	0.79	0.58	0.21
Median concordance correlation	0.79	0.57	0.22
Std dev of concordance correlation	0.01	0.03	(0.02)
Mean Pearson correlation	0.89	0.76	0.13
Median Pearson correlation	0.89	0.76	0.13
Std dev of Pearson correlation	0.01	0.02	(0.01)
Mean Spearman correlation	0.89	0.78	0.10
Median Spearman correlation	0.89	0.78	0.10
Std dev of Spearman correlation	0.01	0.01	(0.01)
Mean Run-pair MARD (%)	8.18	14.91	(6.73)
Median run-pair MARD (%)	8.13	14.84	(6.71)
Std dev of run-pair MARD (%)	0.33	0.70	(0.37)
Document-level metrics			
Documents with identical length (%)	0.05	0.10	(0.05)
Mean document-wise MARD (%)	8.18	14.91	(6.73)
Median document-wise MARD (%)	7.83	13.56	(5.72)
Std dev of document-wise MARD (%)	3.52	8.50	(4.99)

Note: See Appendix B for metric definitions.

Table B9 Consistency Results for Tone of Q&A Answers

Panel A: Tone measured by FinBERT

Metric	GPT-4o-mini	GPT-3.5-turbo	Difference
Sample Size			
Number of documents	2,000	2,000	
Number of runs	50	50	
Overall Inter-rater Agreements			
Fleiss' Kappa	0.66	0.65	0.01
Krippendorff's Alpha	0.66	0.65	0.01
Run-level Agreements			
Mean run-pair Cohen's Kappa	0.66	0.65	0.01***
Mean run-pair agreement (%)	90.64	85.34	5.30***
Document-level Agreements			
Percentage of perfect agreement (%)	67.00	49.55	17.45
Mean document-wise agreement (%)	90.64	85.34	5.30***
Mean majority class strength (%)	93.40	89.67	3.73***
Mean classification uncertainty	0.45	0.46	-0.01
Class-specific Agreements			
Positive (%)	93.26	89.02	4.24
Neutral (%)	81.18	81.22	-0.04
Negative (%)	79.00	76.97	2.03
Class Distribution			
Positive (%)	85.70	73.65	12.05
Neutral (%)	9.45	22.60	-13.15
Negative (%)	4.85	3.75	1.10

Note: See Table B1 for additional notes, and see Appendix B for metric definitions.

Table B9 Consistency Results for Tone of Q&A Answers (Continued)

Panel B: Tone Measured by L&M Dictionary

Metric	GPT-4o-mini	GPT-3.5-turbo	Difference
Sample Size			
Number of documents	2,000	2,000	
Number of runs	50	50	
Overall Inter-rater Agreements			
Fleiss' Kappa	0.57	0.56	0.00
Krippendorff's Alpha	0.57	0.56	0.00
Run-level Agreements			
Mean run-pair Cohen's Kappa	0.57	0.56	0.00**
Mean run-pair agreement (%)	81.98	77.48	4.50***
Document-level Agreements			
Percentage of perfect agreement (%)	44.10	33.70	10.40
Mean document-wise agreement (%)	81.98	77.48	4.50***
Mean majority class strength (%)	87.30	83.85	3.45***
Mean classification uncertainty	0.54	0.56	-0.02
Class-specific Agreements			
Positive (%)	86.96	83.21	3.75
Neutral (%)	76.83	75.66	1.17
Negative (%)	73.49	71.52	1.96
Class Distribution			
Positive (%)	78.40	68.35	10.05
Neutral (%)	7.55	14.95	-7.40
Negative (%)	14.05	16.70	-2.65

Note: See Table B1 for additional notes, and see Appendix B for metric definitions.

Table B10 Consistency Results for Directional Predictions of Future Earnings

Metric	GPT-4o	GPT-4o-mini	GPT-3.5-turbo
Sample Size			
Number of documents	1,000	1,000	1,000
Number of runs	50	50	50
Overall Inter-rater Agreements			
Fleiss' Kappa	0.98	0.99	0.97
Krippendorff's Alpha	0.98	0.99	0.97
Run-level Agreements			
Mean run-pair Cohen's Kappa	0.98	0.99	0.97
Mean run-pair agreement (%)	99.19	99.33	98.54
Document-level Agreements			
Percentage of perfect agreement (%)	94.90	97.30	93.70
Mean document-wise agreement (%)	99.19	99.33	98.54
Mean majority class strength (%)	99.49	99.51	99.05
Mean classification uncertainty	0.27	0.39	0.37
Class-specific Agreements			
Increase (%)	99.09	98.95	98.32
Decrease (%)	98.95	99.13	98.10
Class Distribution			
Increase (%)	54.20	45.10	52.80
Decrease (%)	45.80	54.90	47.20

Note: See Table B1 for additional notes, and see Appendix B for metric definitions.

Table B11 Consistency Results for Point Estimates of Future Earnings

Metric	GPT-4o (1)	GPT-4o-mini (2)	GPT-3.5-turbo (3)
Run-level metrics			
ICC2	0.997	0.864	0.85
Mean concordance correlation	0.995	0.759	0.75
Median concordance correlation	1.000	0.795	0.97
Std dev of concordance correlation	0.007	0.149	0.28
Mean Pearson correlation	0.997	0.868	0.88
Median Pearson correlation	1.000	0.892	0.99
Std dev of Pearson correlation	0.004	0.087	0.14
Mean Spearman correlation	0.999	0.971	0.97
Median Spearman correlation	0.999	0.971	0.97
Std dev of Spearman correlation	0.001	0.007	0.01
Mean Run-pair MARD (%)	2.053	6.176	9.89
Median run-pair MARD (%)	2.020	6.058	9.57
Std dev of run-pair MARD (%)	0.381	1.107	1.66
Document-level metrics			
Documents with identical point estimate (%)	29.200	32.800	10.90
Mean document-wise MARD (%)	2.053	6.421	9.89
Median document-wise MARD (%)	0.123	0.375	0.00
Std dev of document-wise MARD (%)	7.746	18.104	20.69

Note: See Table B1 for additional notes, and see Appendix B for metric definitions.