Can ChatGPT Generate Stock Tickers to Buy and Sell for Day Trading?

Sangheum Cho[†]

First Draft: March 14, 2024

Current Version: April 5, 2024

Abstract

This paper examines the generative feature of ChatGPT for empirical asset pricing. I show that ChatGPT can generate stock tickers that provide a profitable day trading strategy. Using input prompts as multiple Twitter posts, including both macro and firm-specific news by major news providers, I ask ChatGPT to generate lists of stock tickers to buy and sell. The trading strategy based on the buy and sell lists earns significant long-short returns in open-to-close intraday trading. By asking again about the reason for generating those stock tickers, keywords of ChatGPT's answer suggest that tech stocks are important for generating the buy lists, whereas sector- or industry-level analysis is important for generating the sell lists. In particular, ChatGPT's buy and sell lists consist of economically linked stocks through the supply chain, resulting in lower industry concentration than those of their matching groups. The performance is attributable to the stock selection within each industry, the short leg of the strategy, and stronger in the difficult-to-arbitrage stocks, implying that ChatGPT signals' applicability of extracting mispricing signals in text data. As most of the Twitter data consists of non-firm-specific news, this finding sheds light on the literature by showing that ChatGPT can process a bulk of seemingly non-firm-specific news to generate firm-specific mispricing signals.

JEL classification: G10, G11, G12, G14

Keywords: ChatGPT, Empirical Asset Pricing

[†] Development Research Group, The World Bank, 1818 H St NW, Washington, DC, USA,
e-mail: sangheumcho24@gmail.com
I am grateful to Jihoon Goh, Yun-Soo Kim, and Taesoo Nam for valuable comments, and Soobeom Ahn, Sukhee Kang, and Hyeonseong Kim of Financial Modeling Team of Quantit for helpful discussions. The findings, interpretations, and conclusions expressed in this paper are entirely those of the authors and do no reflect the views of the World Bank, its executive directors or the countries they represent.

Electronic copy available at: https://ssrn.com/abstract=4759311

1. Introduction

Large Language Model (LLM)-based generative Artificial Intelligence (AI) models receive significant attention due to their excellence in many tasks. In empirical asset pricing, many papers show the promising role of ChatGPT in predicting future stock returns or producing useful information for investments.[1] The focus of the previous approach is to summarize firm-specific input text data to provide firm-specific signals, such as up and down signals. This is a way to leverage the benefit of LLM to process text data. However, considering another core functionality of LLM-based generative AI, which is to *generate* outputs, it is equally important to test the applicability of generating outputs for empirical asset pricing research. My research question is whether ChatGPT, essentially a *generative* AI model, can *generate* stock tickers to buy and sell when a bulk of information is given. As a methodology to generate multiple stock tickers, I provide not only firm-specific information but also seemingly non-firm-specific news together, whereas the prior literature focuses on a particular stock's news to generate a signal for the stock.

In this paper, I find that ChatGPT can produce a profitable day trading strategy by generating stock tickers to buy and sell for day trading. Using input prompts as multiple Twitter posts, including both macro and firm-specific news by major news providers, I ask ChatGPT to generate a list of stock tickers to buy and sell based on the bunch of input news. The trading strategy based on ChatGPT's buy and sell lists earns significant long-short returns in open-to-close intraday trading. The performance is not attributable to some stocks that are directly mentioned in the input news articles, which suggests that the *generative* feature of ChatGPT works to select relevant stocks by analyzing input news articles. Furthermore, by asking again about the reasons for selecting those lists, the top keywords for selecting the buy lists represent tech stocks, whereas the top keywords for generating the sell lists are related to the sector- or industry-specific considerations. Further analysis shows that the generative feature of ChatGPT selects stocks that are economically linked through supply chains, resulting in lower industry concentration in each of the buy and sell portfolios than their matching groups. The lower level of industry concentration is also consistent with additional results of finding ChatGPT's superior stock selection performance within each industry.

The performance is mostly attributable to the short leg of the strategy, implying that ChatGPT signals' applicability of extracting mispricing signals in text data. As most of the Twitter data consists of

---

[1] See, e.g., Lopez-Lira and Tang (2023), Kim, Muhn, and Nikolaev (2023a), Kim, Muhn, and Nikolaev (2023b), Bond, Klok, and Zhu (2023), Li, Mai, Shen, Yang, and Zhang (2023), Li, Tu, and Zhou (2023), Chen, Tang, Zhou, and Zhu (2023), Glasserman and Lin (2023), Jha, Qian, Weber, and Yang (2024), Pelster and Val (2024), Beckmann, Beckmeyer, Filippou, Menze, and Zhou (2024).

Electronic copy available at: https://ssrn.com/abstract=4759311

non-firm-specific news, this result is new to the literature, showing that ChatGPT can process the bulk of seemingly non-firm-specific news to generate firm-specific mispricing signals.

There are important reasons why I focus on day trading strategy using text data. What machine learning models learn from past data is patterns. Intraday or short-horizon stock return patterns are well-documented areas (e.g., Heston, Korajczyk, and Sadka (2010), Bogousslavsky (2016)). It suggests that machine learning models can be superior in exploiting intraday or short-horizon profitable patterns (e.g., Bogousslavsky, Fos, and Muravyev (2022)). The important thing is that these patterns are not only documented in academic papers but also get a lot of attention in the media and public. Therefore, it is natural to think that text data used as ChatGPT's training data would have lots of descriptions about intraday or short-horizon return patterns. Therefore, this paper focuses particularly on day trading strategy to fully maximize the asset pricing perspectives of the input text data for LLM models.

For the empirical analysis, I use Twitter (currently, X) text data as input prompts.[2] Specifically, I use Twitter posts (i.e., tweets) posted by Bloomberg and the Wall Street Journal. Twitter data is a suitable input variable for ChatGPT due to its conciseness. I use ChatGPT-3.5-turbo.[3] ChatGPT was released at the end of November 2022. My sample period is from December 2022 to December 2023. Therefore, my test data is strictly out-of-sample for the ChatGPT's training set. I collect 222,659 tweets from Refinitiv Eikon News Monitor posted by Bloomberg and the Wall Street Journal. The advantage of using Refinitiv Eikon News Monitor is that the platform provides a Refinitiv Instrument Code (RIC) for some tweets that are directly related to some firms. The number of tweets with firm-specific information is 16,359. This indicates that most of my text data is not directly about firm-specific information. This is a major distinction from previous research using ChatGPT to provide firm-specific signals using firm-specific text information.[4]

Using prompts specifying ChatGPT to pretend to be a *day trader* and to generate stock tickers to buy and sell for day trading, I show that the trading strategy of buying stocks in the buy list and selling stocks in the sell list earns 0.546% per month after adjusting the market factor. Some stocks can be both in buy and sell lists. By strengthening the signals by netting the buy and sell signals, the strategy based on the improved signals earns 3.717% per month. Moreover, the profits mostly come from the short leg and stock having higher limits-to-arbitrage. It suggests that ChatGPT can extract mispricing-related signals in news data.

---

[2] In this study, I will use Twitter and X, interchangeably. A tweet means a post in Twitter.

[3] https://platform.openai.com/docs/models/gpt-3-5-turbo

[4] See, e.g., news headline about a firm (Lopez-Lira and Tang (2023)), MD&A and earnings conferences (Kim et al. (2023b)).

Electronic copy available at: https://ssrn.com/abstract=4759311

The trading strategy has two notable patterns. First, the short leg is correlated with substantial retail investors' overnight buying pressure. It induces a significant overnight reversal of the strategy. The overnight reversal does not disappear as I ask ChatGPT to pretend to be a *swing trader*.[5] It brings concerns about ChatGPT's ability to account for overnight risk management. Second, the trading strategy earns profits using multiple days of lagged signals. This is attributable to the persistence of input news articles, which makes ChatGPT signals to be persistent.

The main contribution of this paper is to show ChatGPT's capability of processing mostly non-firm-specific news to generate firm-specific signals related to mispricing. This finding suggests that generative AIs can successfully link broad-based macroeconomics and industry-wide information to firm-specific content. From the perspective of empirical asset pricing, it implies the potential of LLM-based generative AI to infer both economy-wide latent factor structures and individual stocks' exposure to the innovations in factors. In the canonical high-dimensional data-driven or machine learning methods to estimate factor structures and the corresponding expected returns (e.g., Kelly, Pruitt, and Su (2019) Gu, Kelly, and Xiu (2020)), economy-wide factor structure is proxied by macroeconomic variables or characteristics-managed portfolios. Using them, firm-specific beta is estimated. My approach to using ChatGPT resembles this kind of approach, which shows the potential of making factor models using LLM-based generative AIs.

This paper contributes to the burgeoning literature on applying LLM and generative AI in empirical asset pricing studies. Lopez-Lira and Tang (2023) is a prominent study showing that ChatGPT can process firm-specific news headlines to produce firm-specific signals. Kim et al. (2023b) use individual firm-level MD&A section and earnings conference text data to help investors make decisions. Bai, Boyson, Cao, Liu, and Wan (2023) suggest that the difference between ChatGPT's answers to questions from a firm's earnings conference and human experts' answers has predictive power for future returns. Jha et al. (2024) also focus on conference calls. The main difference between those approaches and my approach is that I use a whole bunch of text data, which is not necessarily firm-specific information. I shed light on the literature by showing that a new way of prompts, which uses all available information, can generate mispricing-related signals.

This paper is closely related to the literature on portfolio selection using ChatGPT. Romanko, Narayan, and Kwon (2023) show that ChatGPT can generate a subset of the S&P 500 index. There are several distinctions between their approach and mine. First, their approach is about the portfolio selection

---

[5] Swing trader refers to traders who hold stocks for several days to weeks. Detailed discussion can be found in the following link: https://www.investopedia.com/terms/s/swingtrading.asp

Electronic copy available at: https://ssrn.com/abstract=4759311

problem, while my approach is to extract cross-sectional mispricing signals. Second, they do not use text data as input variables, which limits their universe as the subset of the S&P 500 index. I have a broader cross-section of stocks to form portfolios for long-short strategies based on not only firm-specific but also economy-wide news.

This paper is also closely related to the literature examining the economic interpretability of machine learning, deep learning, and AI models. Due to the black-box-like complex model architecture of AI models, it is important to the economic mechanism of their output. Karolyi and Van Nieuwerburgh (2020) address the importance of understanding the economic meaning of outputs from machine learning models in empirical asset pricing. Avramov, Cheng, and Metzker (2023) suggest that machine learning models' signal is concentrated in hard-to-arbitrage stocks. Byun, Cho, and Kim (2024) present that the deep learning models' performance is related to mispricing signals attributable to behavioral biases. This paper suggests that the ChatGPT provides outcomes by considering sector- or industry-level information and linking it to generate individual stock's ticker. Furthermore, I also suggest that economic links, such as supply chain linkage, have certain roles for ChatGPT in generating relevant stock tickers.

This paper is also closely related to the literature using various news data sources as input prompts for ChatGPT. Vamossy and Skog (2023) use StockTwits data as input variables for LLMs to extract sentiment. Bond et al. (2023) use business news to construct market-wide sentiment. I use Twitter news data posted by major news providers.

This paper is also related to the active discussion about the hallucination of generative AIs.[6] As I iterate 30 times to ask ChatGPT to generate lists of stocks to buy and sell, I can construct buy and sell portfolios holding stocks that are similar to the usual number of stocks used in decile portfolios in empirical asset pricing literature. It suggests some potential avenues for generative AI to overcome hallucination issues when proper inputs and prompts are given.

The remainder of this paper is as follows. Section 2 describes Twitter news data and ChatGPT prompt engineering using the text data. Section 3 discusses the portfolio construction using ChatGPT day trading signals. Section 4 provides empirical results. Section 5 concludes.

---

[6] See, e.g., Ji, Lee, Frieske, Yu, Su, Xu, Ishii, Bang, Madotto, and Fung (2023).

Electronic copy available at: https://ssrn.com/abstract=4759311

2. Twitter News Data and ChatGPT Prompt Engineering

2.1. Twitter Data

Twitter data is a well-suited text data for my analysis due to its conciseness. The total number of words is limited to 280 characters, which is between 40 to 70 words.[7] Therefore, news providers' Twitter posts should be concise and have well-summarized information.

I collect Twitter raw text data from Refinitiv Eikon News Monitor by searching tweets posted by Bloomberg (@business), Bloomberg Markets (@markets), Bloomberg Economics (@economics), Bloomberg Opinion (@opinion), and the Wall Street Journal (@WSJ).[8]

The initial number of tweets by Bloomberg and the Wall Street Journal is 305,644 during my sample period covering December 2022 to December 2023. Bloomberg (@buisness) posted 159,053 tweets, Bloomberg Markets (@markets) posted 44,740, Bloomberg Economics (@economics) posted 47,052, Bloomberg Opinion (@opinion) posted 15,096, and the Wall Street Journal (@WSJ) posted 39,703 tweets.

2.1.1. Examples

Table 1 shows some examples.[9] The first example shows that Bloomberg (@business) posted a tweet on January 18, 2023, 8:47:56 AM in the Eastern Standard Time (EST).[10] The tweet is related to Microsoft. This firm-specific tag is provided by Refinitiv Eikon News Monitor. Refinitiv Eikon News Monitor provides a Refinitiv Instrument Code (RIC) for a tweet when it contains some firm-specific information. The column labeled as RIC indicates RIC information.

Regarding the timing of the tweets posted, I classify them into *Overnight News* and *Intraday News*. A tweet is classified as *Overnight News* when it is posted from the previous trading day's market close (4:00 PM) to the right before today's market opening (9:30 AM). A tweet is classified as *Intraday News*

---

[7] https://developer.twitter.com/en/docs/counting-characters

[8] In this paper, unless otherwise noted, Bloomberg includes @business, @economics (Bloomberg Economics), @markets (Bloomberg Markets) and @opinion (Bloomberg Opinion).

[9] Appendix Figure 1 and Appendix Figure 2 show the screenshots of the Twitter page and the corresponding Bloomberg page for each case discussed in Table 1.

[10] Each tweet indicates posted time based on the time zone of the geographic location of access.

Electronic copy available at: https://ssrn.com/abstract=4759311

when it was posted during the market opening from 9:30 AM to 4:00 PM. In this example, it is classified as *Overnight News* since it was posted at 8:47:56 AM.

In terms of predicting future returns, *Overnight News* is used to predict the same day's intraday returns, whereas *Intraday News* is used to predict the next trading day's intraday returns. Therefore, in this example, this news will be used as a part of input prompts for ChatGPT to predict the upcoming intraday return on January 18, 2023.

On the other hand, some tweets can be posted during the intraday trading sessions. The second example is for Bed Bath & Beyond and was posted on February 7, 2023, at 9:34:42 AM EST. Since the regular market opens at 9:30 AM EST, this post is classified as *Intraday News*. This tweet is used to predict the next trading day's intraday return, which will be realized on February 8, 2023.

However, most tweets do not have RIC information. The third example shows that Bloomberg (@opinion) posted a tweet talking about the upcoming CPI news release. The tweet delivers expectations about the CPI data, which will be released at 8:30 AM EST for each scheduled day. A few minutes later, Bloomberg posted several updates on CPI data as CPI data was formally released. For example, the fourth tweet released at 8:32:39 AM is about the headline CPI and core CPI, whereas the fifth tweet at 8:37:06 AM is about energy price. These tweets are used to predict the same day's intraday returns as they are classified as *Overnight News*. The last example shows a labor market-related tweet posted intraday so that it is used to predict the intraday return of the next day.

2.1.2. Cleaning Twitter Data

The sample period is from December 2022 to December 2023. The raw Twitter data consists of tweets posted by Bloomberg and the Wall Street Journal, which is 305,644 tweets during the sample period. There are 159,053 tweets by Bloomberg (@buisness), 44,740 by Bloomberg Markets (@markets), 47,052 by Bloomberg Economics (@economics), 15,096 by Bloomberg Opinion (@opinion), and 39,703 by the Wall Street Journal (@WSJ).

I follow several steps to clean the text data. I remove the internet hyperlink information from each tweet. I also remove the Twitter poster information starting from the @ sign, such as @business. Some of the tweets have identical information. This is because the poster repeatedly posted tweets for various purposes. In this case, I remain the last tweets in each intraday and overnight period. This procedure gives

Electronic copy available at: https://ssrn.com/abstract=4759311

me 222,659 tweets for the main analysis. The daily average number of tweets is 795. In the *Overnight News*, there are, on average, 570 tweets, whereas 225 are in *Intraday News*. There are, on average, 24 words per tweet. The number of tweets with RIC information is 16,359. *Firm-Specific News* indicates tweets having RIC information.

Figure 1 shows the word cloud for my tweet sample. Panel A shows a word cloud using all news. Some keywords include US, China, Interest Rates, and many others. Panel B shows a word cloud using only *Firm-Specific News*. The figure shows keywords such as Company, Amazon, Microsoft, Tesla, Goldman Sachs, and many others. On the other hand, Panel C shows a word cloud excluding *Firm-Specific News*. The keywords are similar to Panel A. It may be attributable to the fact that only around 7% of tweets are *Firm-Specific News*. Therefore, my news text data is mostly about *non-Firm-Specific* news, which makes my research different from Lopez-Lira and Tang (2023) and many others using firm-specific text data to generate some signals from ChatGPT.

2.2. ChatGPT Prompt

My approach to using ChatGPT differs from the previous approach in two ways. First, I focus on the role of *generating* outputs of generative AIs. A plausible output to test this applicability for empirical asset pricing would be a list of stocks. For this purpose, I provide multiple news articles, including both firm-specific and macro news articles, which are, on average, 795 tweets per day. This is a major difference from my approach compared to the previous approach, which provides ChatGPT with news articles focusing on a particular stock. Second, I focus particularly on day trading strategy. This is to fully maximize the abundant text data describing intraday patterns that potentially appear in ChatGPT's training data.

Therefore, I provide a prompt specifying ChatGPT to trade as a professional day trader by analyzing multiple news articles to generate at least 100 stocks to buy and sell.[11] Specifically, I use the following prompts to generate signals.

---

[11]

Electronic copy available at: https://ssrn.com/abstract=4759311

Forget all of your previous instructions. Pretend to be a financial expert. You are a professional day trader.

You trade stocks based on news to implement the “News Trading” strategy. You analyze news articles by Bloomberg and the Wall Street Journal posted on Twitter. To make money, you pick some US-listed individual stocks to buy or sell in the short term.

Your answer should be the following JSON format:

BUY:[TICKER1,TICKER2]

SELL:[TICKER3,TICKER4]

Your universe consists of US-listed individual stocks that can be traded on major exchanges such as the NYSE, NASDAQ, and AMEX.

Each buy and sell list should have at least 100 US individual stocks’ ticker symbol. For the buy list, the most likely to go up comes first, and for the sell list, the most likely to go down comes first.

Let’s begin. Pick stocks to buy and sell by analyzing the following news articles: **NEWS_COMES_HERE**

There are technical details to obtain high-quality responses. The length of news should be limited by considering the maximum number of tokens that ChatGPT can process at a time. I use ChatGPT-3.5-turbo-16k for my analysis. The token limit per API request for the model is 16,385 tokens, including both input prompt and output ticker lists. The average number of tokens per tweet is around 30. Therefore, by a simple calculation, the maximum number of tweets provided to ChatGPT should not exceed 546 (16,385/30). As the average number of tweets per day is 795, I should divide full one-day Twitter posts into several batches. For this purpose, I make distinct batches for each *Intraday News* and *Overnight News*. Fortunately, since *Intraday News* has on average 6,920 tokens, none of the *Intraday News* batch exceeds the maximum number of tokens. However, for the *Overnight News*, the average number of tweets is 570, which is still on average greater than the maximum number of tweets that ChatGPT can process at a time. Therefore, I divide *Overnight News* into several batches. More specifically, by starting the most recent

Electronic copy available at: https://ssrn.com/abstract=4759311

tweets, I sum up the number of tokens until it reaches 14,000. Once the cumulative number of tokens exceeds 14,000, then I make a new batch. This process gives me on average 12,276 tokens for each batch in *Overnight News*. The average number of batches in *Overnight News* is 1.4.

2.3. Output Examples

I report several examples of output responses from ChatGPT using tweets in Table 1. For the first tweet in Table 1, which is about Microsoft, the total number of tweets in the batch that the tweet is included is 448. This batch covers news from January 17, 6:05:11 PM to January 18, 9:28:14. The number of tokens the batch has is 14,408.

Here is the real output from the prompt.

> *BUY*: [**AAPL, GOOGL, AMZN, MSFT, TSLA, FB, NVDA, JNJ, BRK-B, JPM, V, PG, UNH, MA, INTC, ADBE, HD, VZ, DIS, NFLX, PYPL, CRM, XOM, CMCSA, CSCO, PEP, ABBV, ABT, BAC, KO, T, MRK, NKE, NVAX, TMO, WMT, CVX, PFE, IBM, ACN, COST, BMY, AMD, GOOG, TMUS**, NSRGY, **DHR**, **LOW, NEE, MMM, QCOM, ORCL, UPS, AMGN, DE, SO, LIN, BA, GS, COP, BHP, DHR**, **CAT, EL, NOW, TT, SPOT, DOCU, ZM, CSCO, WDAY, MU, PTON, COIN**]
>
> *SELL*: [**AMC, GME, BB, ACB, PLUG, VXRT, RIOT, MVIS, SPCE, RBLX, COIN, TIGR, WKHS**, SPAQ, CCIV, **ZEV, JAGX, XPEV, FUBO, OCGN, NIO, TSLA**, ARA, **APH, MAXR, CGC, FUV, QS,** RMO, **ABNB, NKLA, TLRY, CBAT, DM, FSLY,** SMFR, IPOF, **ALPP, LCID, GOEV, FCEL, RIDE, FSLR,** CLII, **CAN, CPNG, DADA,** DIDI, **NNDM, LMND, ROOT, NFLX**, TWTR, **SQ, PTON**, **NET, HOOD, CLOV**, **MTTR, ROKU, DKNG, CRWD, BILL, DDOG, OPEN, UPST**]

The *BUY* list includes MSFT, which is the ticker symbol for Microsoft. Besides Microsoft, I highlight ticker symbols that can be found in CRSP ticker symbols as of January 18, 2023. Out of the total 74 tickers that are generated by ChatGPT, 73 tickers are real tickers. The *SELL* list also provides a substantial number of tickers that exist in the real world. Out of the total 66 tickers generated, 57 are indeed traded in the real-world stock market.

Electronic copy available at: https://ssrn.com/abstract=4759311

The lists should be dealt with carefully because several tickers appear both in *BUY* and *SELL* lists. For example, TSLA appears in both *BUY* and *SELL* lists. In this case, TSLA can be considered to have a net zero signal. The multiple iterations of generating lists may strengthen signals. In this paper, I repeat 30 times of asking the prompt for each batch of tweets. The 30 iterations are determined to have at least 10% of stocks have non-zero net buy and net sell signals.

With the 30 iterations of producing each buy and sell list, MSFT is selected 26 times in the *BUY* list and 12 times in the *SELL* list. Accordingly, by netting the *BUY* and *SELL* signals, 14 can be considered as a net buy (*NET_BUY*) signal. Besides the batch of this example, there are one more *Overnight News* batch and one more *Intraday News* batch. Since I repeat 30 times for each of the three batches, the maximum number of *BUY* or *SELL* signals is 90. The three batches of tweets have a total of 728 tweets with 25,069 tokens.[12] In total, MSFT appears 80 times in *BUY* lists and 20 times in *SELL* lists, resulting in a 60 *NET_BUY* signal.

The second example is the second tweet in Table 1. It shows negative sentiments for Bed Bath & Beyond (BBBY). The tweet is included in one of three batches from February 7, 2023, 9:33:08 AM to February 8, 2023, 9:28:28 AM, with 851 tweets and 27,279 tokens. The following is a real example of output containing the tweet about BBBY.

> *BUY*: [**ETSY**, **AMZN, AAPL, MSFT, GOOGL, FB, PYPL, ADBE, SHOP, NFLX, TSLA, CRM, NVDA, INTU, AMD, COST, MA, DIS**, **SBUX, WMT, HD, LULU**, **ORCL, BA, ADSK, NVAX, GILD, MRNA, PFE, MRK, JNJ, CL, PG, AAP, KMB, FDX, UPS, DAL, AAL, UAL, DAL, LUV, CCL, NCLH, RCL, MAR, HLT, WH, M, JWN**, LB, **GPS, KSS, TJX, ROST**, **LVS, MGM, CZR, WYNN, MCD, YUM, CMG, QSR, SBUX, KKR, GS, MS, JPM, C, AAL, UAL, DAL**, **AAL, SPG, PLD, EQIX, AMT, CCI, PLTR, SNOW, SQ, ZM, PLUG, BE, BLDP, QS**, XL, **FCEL**, SBE, **NIO, LI, XPEV, GM, F, RIVN, SNAP, GDRX, FIGS**]

---

[12] Note that the maximum number of ChatGPT to process at single request is 16,385. As the total number of tokens is 25,069 tokens, this example provides the reason why I need to divide news into several batches.

Electronic copy available at: https://ssrn.com/abstract=4759311

SELL: [**BBBY**, ETH, **LOVE, FLGT**, **ZUO, IAC, EXAS**, **PLCE, YALA, AN**, **PLBY, OWLT, FULC, UFO, RAIN, GOLF, CLOV,** HMBL, **AMC, GME, BBIG,** SPRT, **IRNT**, SELA, **CRVS, MVIS, WEED, TIRX, VERB, RUBY**, BNA, **FUBO**, DCRC, **NLSP**, **RWLK**, **SOLO**, PROG, **IRNT**, **LCID**, **BYND**, **RIOT, TSLA**, **MU, FLUX**, RICE, **TMC**, AAVE, **SOL**, CRV, BNB, DOGE, SHIB, MATIC]

We can see that BBBY is selected as the first ticker for the *SELL* list. Overall, in the *BUY* list, 95 tickers out of 98 tickers are matched with the CRSP tickers, and in the *SELL* list, 39 tickers out of 53 tickers. With the 30 iterations for each of the three batches of tweets, BBBY is selected for the *BUY* list just 2 times but 47 times for the *SELL* list, resulting in a 45 times net sell (*NET_SELL)* signal.

The outputs in these two examples are quite straightforward since they include firm-specific information. However, 93% of my tweets data is non-firm-specific. Then, which kinds of stock tickers are generated on days with important macroeconomic news days rather than firm-specific information?

The third example in Table 1 shows the macro news case. November 14, 2023, is the day CPI news was released, which was an important topic for stock market investors throughout 2023. The tweet and the following tweets suggest that CPI has not increased much. I present the five stocks selected most frequently in the *BUY* and *SELL* lists, respectively.

BUY: [MSFT, NVDA, GOOGL, AMZN, PYPL, …]

MSFT is selected for 45 times, NVDA for 44, GOOGL for 43, AMZN for 42, and PYPL for 41. The news about easing inflation may induce positive sentiment for investors, especially for tech stock investors. This concept might be a brief answer for why tech stocks are frequently selected as the *BUY* list on the CPI release day.

The *SELL* list looks like the following:

SELL: [HAL, NCLH, RCL, UAL, M, …]

HAL is selected for 14 times, NCLH is for 12, RCL is for 12, UAL is for 11, and M is for 11. The *SELL* list is more nuanced than the *BUY* list because none of the tickers have the RIC information. HAL is a ticker symbol for Halliburton Company, which operates energy businesses worldwide. The fifth example in Table 1 says that energy prices are falling, which can negatively affect the profitability of energy companies. Therefore, HAL can be selected for the *SELL* list. However, NCLH is the Norwegian Cruise

Electronic copy available at: https://ssrn.com/abstract=4759311

Line Holdings Ltd., which has no clear reason for the selection. One interesting observation is that RCL is also a cruise company, Royal Caribbean Cruises Ltd. It seems that the generative characteristics of ChatGPT may generate stock tickers sharing similar industry exposures, which are likely to appear together in its training text data.

3. ChatGPT Signals and Portfolio Construction

This section discusses the construction of various ChatGPT signals, portfolio formation using the signals, and theoretical asset pricing considerations of this approach.

3.1. Construction of *BUY* and *SELL* Signals

3.1.1. ChatGPT-based *BUY* and *SELL* List

The *BUY* and *SELL* signals are constructed from ChatGPT by asking to pick stocks to buy and sell based on tweets posted on Twitter by Bloomberg and the Wall Street Journal. For each day, I provide multiple tweets posted from the previous 24 hours before the market opening of the day, which is 9:30 AM US Eastern Standard Time (EST). Due to the maximum number of tokens that ChatGPT can process per API request, I construct several batches of tweets to ensure the input tweets do not exceed the limit, which is 16,385 tokens, including both prompts and responses. We repeat it 30 times per batch, which produces 30 lists of *BUY* and *SELL* stocks per batch of tweets. The reason for choosing 30 iterations is to ensure that the number of stocks with the main signals is around 10% of the cross-section of stocks. It is to make my portfolios have a similar number of stocks to the usual decile portfolios discussed in the empirical asset pricing literature.

3.1.2. Baseline *BUY* and *SELL* Signals

Using *BUY* and *SELL* lists for each day, I construct stock-day-level *BUY* and *SELL* signals. A stock has a *BUY* signal of 1 when it appears at least once in the *BUY* lists; otherwise, it is 0. The *SELL* signal is similarly constructed using the *SELL* lists. Therefore, our *BUY* and *SELL* signals are dummy variables.

Electronic copy available at: https://ssrn.com/abstract=4759311

3.1.3. *SUM_BUY* and *SUM_SELL* Signals

SUM_BUY and *SUM_SELL* signals are the sum of selections. For example, if a stock appears 15 times in the *BUY* list, then the *SUM_BUY* signal is 15. Similarly, if a stock appears 10 times in the *SELL* list, then the *SUM_SELL* signal is 10.

I iterate 30 times to generate the *BUY* and *SELL* lists for each batch of tweets. On the other hand, since a day can have multiple number of batches, the total number of selections for a ticker can be greater than 30. For example, a day can have three batches. If MSFT is selected 20 times for *BUY* lists for the first batch, 15 times for the second batch, and 25 times for the last batch, then the *SUM_BUY* signal is 60. Similarly, if MSFT appears in *SELL* lists 5 times, 5 times, and 10 times for each of the three batches, then the *SUM_SELL* signal is 20.

3.1.4. *NET_BUY* and *NET_SELL* Signals

NET_BUY and *NET_SELL* signals are obtained by taking the difference between *SUM_BUY* and *SUM_SELL* signals. If a stock has a *SUM_BUY* signal greater than its *SUM_SELL* signal, then the stock has a positive *NET_BUY* signal, which is the difference between *SUM_BUY* and *SUM_SELL* signals. *NET_SELL* signal is similarly constructed for stocks having a *SUM_SELL* signal greater than their *SUM_BUY* signals by taking their differences. By construction, *NET_BUY* and *NET_SELL* signals are positive.

For example, if MSFT has a *SUM_BUY* signal of 60 and a *SUM_SELL* signal of 20, then MSFT has a *NET_BUY* signal of 40. In the case of BBBY in my previous example, which shows that BBBY is selected 2 times for the *BUY* list and 47 times for the *SELL* list, then BBBY has *NET_SELL* signal as 45.

3.1.5. Construction of Daily-level Panel Dataset

The sample period with ChatGPT signals is from December 2022 to December 2023, which is 271 trading days. I merge these stock-day-level signals with the daily CRSP individual stock returns data. Following the prior literature (e.g., Lopez-Lira and Tang (2023)), I use stocks having share codes 10 and 11 that are

Electronic copy available at: https://ssrn.com/abstract=4759311

traded in NYSE, AMEX, and NASDAQ. Furthermore, I impose several restrictions on my sample. I drop observations with zero trading volume and missing the previous day's price or returns. I also use stocks with full trading histories in the previous 5 trading days.

3.2. Summary Statistics of ChatGPT Signals

Table 2 reports summary statistics of ChatGPT day trading signals. Panel A reports the full sample summary statistics of *BUY* and *SELL* signals. The average number of stocks is 3,911. The average of the *BUY* signal is 0.18, which means that 18% of US individual stocks, or 701 stocks, are selected by ChatGPT as *BUY* lists. The number of observations having a *BUY* signal is 190,130. The average of the *SELL* signal is 0.24, implying that 24% of stocks, or 927 stocks, are selected as *SELL* lists. The number of observations having a *SELL* signal is 251,410.

D_TWTR is a dummy variable for RIC information. On average 1% of tweets have RIC information. This is the key difference between my analysis and the previous approach. The previous research focuses on firm-specific information to put into LLM. However, my approach uses mostly non-firm-specific news as an input variable for LLM.

Panel B reports a detailed distribution of *SUM_BUY* and *SUM_SELL* signals and *NET_BUY* and *NET_SELL* signals. For the *SUM_BUY* signal, 190,130 observations have *SUM_BUY* signal, and the average is 6.02. *SUM_SELL* signal has 251,410 non-missing observations, and the average value is 3.15. By comparing their average and median values, *SUM_BUY* and *SUM_SELL* signals have skewed distributions.

NET_BUY and *NET_SELL* signals have fewer observations because they are obtained from netting out *SUM_BUY* and *SUM_SELL* signals. The number of observations for the *NET_BUY* signal is 119,880. The difference between this and the number of observations of *SUM_BUY* signal (190,130) implies that among observations with *SUM_BUY* signals, 70,250 of them also have *SUM_SELL* signals. The average of the *NET_BUY* signal is 5.54. The number of observations for the *NET_SELL* signal is 163,508. The average value of the *NET_SELL* signal is 1.89.

N_TWTR is the number of RIC information for a stock-day-level observation. In the full sample, 8,229 stock-day observations have RIC information in the previous 24 hours of Twitter news data, and the

Electronic copy available at: https://ssrn.com/abstract=4759311

average is 2.72. The total number of unique stocks that appeared in some tweets with RIC information is 735.

3.3. Portfolio Construction based on ChatGPT signals

I construct several portfolios using various ChatGPT signals. Using the baseline *BUY* and *SELL* signals, I construct *BUY* and *SELL* portfolios by equally weighting stocks having *BUY* and *SELL* signals, respectively. Using the *SUM_BUY* and *SUM_SELL* signals and *NET_BUY* and *NET_SELL* signals, I construct some signal-weighted portfolios. The *SUM_BUY* portfolio is constructed by weighting stocks using their *SUM_BUY* signals. The *NET_BUY* portfolio is constructed by weighting stocks using their *NET_BUY* signals. The *BUY* and *SUM_BUY* portfolios use the same cross-section of stocks to construct portfolios, while they use different weighting. However, the *NET_BUY* portfolio uses a smaller sample than the *BUY* and *SUM_BUY* portfolios because the *NET_BUY* signal is constructed from the netting *NET_SELL* signals. The percentage of stocks out of the entire cross-section in the *BUY* and *SUM_BUY* portfolios is 18%, whereas 11.3% in the *NET_BUY* portfolio. For the *NET_SELL* portfolio, 15.4% of stocks are in the *NET_SELL* portfolio.

To examine the strength of signals, I construct long-short portfolios. *BMS* is the difference between *BUY* and *SELL* portfolio returns. *SUM_BMS* is the difference between *SUM_BUY* and *SUM_SELL* portfolio returns. *NET_BMS* is the difference between *NET_BUY* and *NET_SELL* portfolios.

My baseline approach is to hold stocks from the market opening to the market closing because I test the applicability of ChatGPT to help day trading. Therefore, the main portfolio returns are open-to-close price returns. Market returns are proxied by using SPY (SPDR S&P 500 ETF Trust). The market return is also an open-to-close price return. I proxy risk-free rates using open-to-close price returns of BIL (SPDR Bloomberg 1-3 Month T-Bill ETF). The excess return is the difference between portfolio returns and risk-free rates. As a risk-adjusted return, CAPM alpha is estimated using the market factor proxied by the open-to-close price return of SPY return minus BIL return. The standard errors are estimated using the Newey-West adjustment.

Electronic copy available at: https://ssrn.com/abstract=4759311

3.4. Theoretical Consideration

Let firm-specific news is $X_{i,t}$ and macro news is X_t. Prior research uses firm-specific news, $X_{i,t}$, as input text data for ChatGPT to produce a firm-specific signal. Therefore, the return-generating process can be considered as follows:

$$r_{i,t+1} = f\left(X_{i,t}\right) + \epsilon_{i,t+1}.$$

My approach is to use $X_{i,t}$ and X_t together to generate multiple stock-level signals. It makes my approach to be considered as to *generate* a weight vector, w_t for individual stocks' weight, $w_{i,t}$. Furthermore, as I specify my input prompt to "make money," my approach can be considered to obtain the weight vector to maximize investment profits. The following expression characterizes my approach:

$$\text{argmax}_{w_t} \sum_i w_{i,t}\left(X_{i,t}, X_t\right) r_{i,t+1} \tag{1}$$

where $w_{i,t}\left(X_{i,t}, X_t\right)$ is the weight for each stock generated by using input variables as $X_{i,t}$ and X_t.

As the solution of equation (1) with plausible constraints implies a stochastic discount factor, my approach paves the way to use ChatGPT to construct a factor model using both firm-specific and macroeconomic variables or economy-wide variables. Kelly et al. (2019), Gu et al. (2020), Gu, Kelly, and Xiu (2021), and Chen, Pelger, and Zhu (2024) use standard firm characteristics, macro variables, or characteristics-managed portfolios to obtain latent factor models spanning efficient frontiers. My approach is an LLM- and generative AI-based version of those studies.

4. Empirical Results

This section reports the main empirical results. I start by reporting the relationship between the stock-level characteristics and ChatGPT signals. Then, I discuss the performance of portfolios constructed by using ChatGPT signals.

Electronic copy available at: https://ssrn.com/abstract=4759311

4.1. Stock-Level Characteristics and ChatGPT Day Trading Signals

Table 3 shows panel regression results of regressing ChatGPT day trading signals on various stock-level characteristics. All estimations include date-fixed effects. The standard errors are two-way clustered by stock- and day-level. Panel A uses stock-level characteristics as the main independent variables. *SIZE* is the log of market capitalization. *MOM1M* is the previous 20 days of returns. *MOM12M* is the previous 12 to 1-month returns. $RET\ (t-1)$ is the previous day's returns.

Column (1) shows that stocks with larger market capitalization tend to have a *BUY* signal, and their relationship is statistically significant. This is an expected result since ChatGPT learns from past data, which provides more information about larger firms. Columns (2) to (4) show results for past returns. Stocks with momentum in various frequencies tend to be selected in the *BUY* list. However, Column (5) shows that most past returns provide inconsistent direction to the *BUY* signal when *SIZE* is controlled. Column (6) also shows that the *SELL* signal is significantly related to firm size. Columns (7) to (10) also present that *SIZE* might be the most important determinant of the *SELL* signal.

Panel B uses an indicator for RIC information (*D_TWTR*) and various market information as the main independent variables. *D_TWTR* for a stock is a dummy variable having 1 one when the RIC information of the stock is tagged in a tweet, which is provided by Refinitiv Eikon News Monitor. *N_TWTR* for a stock is the total number of tweets in which the stock appears. *D_SP500* is a dummy variable having 1 one a stock is a constituent of the S&P 500 index. *D_NASDAQ* is a dummy variable having 1 one a stock is traded in NASDAQ.

Column (1) shows that when a stock's RIC information is in a tweet, then it has a much higher likelihood of being selected in the *BUY* list. However, Column (2) shows that the intensive margin of appearance of firm-specific information is a less important determinant. Columns (3) and (4) show that S&P 500 index membership and NASDAQ are also significant determinants of the *BUY* signal. Column (5) shows that after controlling *SIZE*, *D_TWTR*, and indicator for the membership for the S&P 500 index and NASDAQ are significant. Columns (6) to (10) suggest similar interpretations while the NASDAQ's sign is flipped. Among the stocks in the S&P 500 index, 59% and 58% have *BUY* and *SELL* signals, respectively.

Panel C uses trading frictions as the main independent variables. *TURN* is the previous trading day's share turnover. *ILLIQ* is the Amihud (2002) illiquidity measure multiplied by 10,000. *BASPREAD* is the bid-ask spread, and *RETVOL* is the previous 20 days of return volatility. Columns (1) to (4) show that in the univariate regressions, stocks having better liquidity and lesser trading frictions tend to be selected

Electronic copy available at: https://ssrn.com/abstract=4759311

as *BUY* list. Column (5) shows that *TURN* is a consistent determinant after controlling *SIZE*. Columns (6) to (10) show similar results that *TURN* is also important for *SELL* signal.

Panel D uses earnings announcement day as the main independent variable. *D_RDQ* is a dummy variable having 1 when a firm announced earnings in the previous 7 days. Column (1) shows that the earnings announcement periods are not an important determinant of the *BUY* signal. Furthermore, when controlled with other control variables, Columns (2) to (5) show that days without earnings announcement would have a higher probability of being in the *BUY* list. Columns (6) to (10) provide similar patterns that earnings announcement days are not the important determinants for the *SELL* signal. This indicates that my approach is fundamentally different from the previous approach in terms of the characteristics of input text data. Appendix Table 1 shows results after including SIC two-digit codes (SIC2) fixed effects. Results are qualitatively similar to using only date-fixed effects.

4.2. Intraday Performance

Table 4 reports the intraday excess returns and CAPM alpha of the day trading strategy based on ChatGPT day trading signals in monthly percentage terms.[13] Panel A reports portfolio returns by the *BUY* and *SELL* signals and their difference, denoted by *BMS*. The *BUY* portfolio obtains -0.098% of monthly returns, and the *SELL* portfolio shows -0.644% of returns. *BMS* return is 0.546%, which is statistically significant. After adjusting risk, both *BUY* and *SELL* portfolios show negative returns. However, *BMS* shows positive CAPM alpha as 0.525%, which is weakly statistically significant. These results show that *BUY* and *SELL* signals have cross-sectional return predictability, but both *BUY* and *SELL* stocks underperform compared to the benchmark.

In Panels B and C, I present stronger results using weighted signals. Panel B reports that the *SUM_BUY* signal earns positive excess returns as 1.585% monthly, and the *SUM_SELL* portfolio shows 0.073% of returns. When I adjust the market factor, the *SUM_BUY* portfolio's return is almost subsumed by the market factor, but the *SUM_SELL* portfolio shows negative statistically significant returns of -1.450%. The *BMS* return is 1.629% and statistically significant. This shows intensive selection by ChatGPT can be considered a predictive signal, especially for the short side of the strategy.

---

[13] For the sake of ease of interpretability and comparability with usual return predictability literature, I report all the portfolio excess returns and CAPM alphas in monthly percentage terms.

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel C reports using *NET_BUY* and *NET_SELL* signals. *NET_BUY* portfolio earns 1.850%, which is even higher than the *SUM_BUY* return, whereas *NET_SELL* earns -1.663% per month. Their difference is substantial, which is 3.513% per month. After risk adjustment, *NET_BUY* shows 0.524% of CAPM alpha, and *NET_SELL* shows -3.193% of significant CAPM alpha. Their long-short return, *NET_BMS*, shows 3.717% of statistically significant risk-adjusted return. These results imply that *NET_BUY* and *NET_SELL* signals are much more accurate signals than *SUM_BUY* and *SUM_SELL* signals by netting out some potential noise.

Figure 2 graphically shows various portfolio returns. Panel A reports for *BUY*, *SELL*, and *BMS* returns, Panel B reports for *SUM_BUY*, *SUM_SELL*, and *SUM_BMS*, and Panel C reports for *NET_BUY*, *NET_SELL*, and *NET_BMS*. Panel C shows the clear distinction between *NET_BUY* and *NET_SELL* portfolio performance. Except for the first half of January 2023, the *NET_BMS* portfolio consistently earns positive returns.

4.3. Effectiveness of Generative Feature of ChatGPT

This paper examines the applicability of the *generative* feature of ChatGPT in empirical asset pricing. Only 7% of the input Twitter news articles have firm-specific information, which results in only 30 stocks per day being directly mentioned in Twitter news articles in my sample period. Moreover, among 442 stocks with the *NET_BUY* signal, only 21 stocks (4.75%) are mentioned in the Twitter news article. Among 603 stocks with the *NET_SELL* signal, only 6 (1.00%) of stocks are mentioned on Twitter. Therefore, the results show that the generative feature of ChatGPT coupled with its characteristics as LLM can successfully handle seemingly non-firm-specific news to find relevant firms.

This section provides supportive evidence for the effectiveness of the generative feature of ChatGPT in implementing day trading. Specifically, this section examines how much the portfolio performance is affected by the Twitter-mentioned firms.

Table 5 reports the results of the intraday performance of the day trading strategy using subsamples of stocks separated by whether or not they are mentioned in Twitter news articles. A stock is classified as Twitter-mentioned when the stock's Refinitiv Instrument Code (RIC) is tagged in some tweets. RIC is provided by Refinitv News Monitor.

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel A shows the results by excluding the Twitter-mentioned firms. The *NET_BUY* portfolio earns 1.946% of monthly excess returns, which is 0.626% of monthly CAPM alpha, whereas the *NET_SELL* portfolio presents -3.241% of monthly excess returns, which is -1.724% of monthly CAPM alpha. The resulting *NET_BMS* exhibits monthly excess returns as 3.670% and monthly CAPM alpha as 3.868%. The magnitude of the long-short strategy is statistically significant and almost quantitatively similar to 3.717% of the *NET_BMS* returns of using all stocks. On the other hand, in Panel B, when portfolios are constructed using only Twitter-mentioned firms, then *NET_BMS* is no longer significant. Therefore, in the cross-section, the portfolio performance is not solely dependent on the Twitter-mentioned firms.

Table 6 presents time-series evidence. This table reports the time-series regression results of examining the relationship between intraday portfolio performance and the intensity of firm-specific information. The main dependent variable is excess portfolio returns. The main independent variable is *D_TWTR_HIGH*, which is a dummy variable having 1 when a day has above the median percentage of firm-specific Twitter news articles.

Column (1) shows that the performance of the *NET_BUY* strategy is not positively associated with days with higher firm-specific news. Column (2) also shows that the performance of the *NET_SELL* strategy is not affected by the days with more firm-specific news. As a result, Column (3) shows that the *NET_BMS* strategy is not affected by more firm-specific news.

As I suggest that the generative feature of ChatGPT can successfully generate stock tickers that are not directly mentioned in Twitter news articles, I provide more direct evidence that ChatGPT even generates stock tickers using news articles having no firm-specific information.

Table 7 reports the results of testing portfolio performance conditional on the inclusion or exclusion of firm-specific news in input text data for ChatGPT. Panel A is the benchmark *NET_BMS* performance of using all news. Panel B is the *NET_BMS* performance based on only firm-specific news to generate ChatGPT signals. However, the performance is rather deteriorated compared to the benchmark. On the other hand, Panel C, which uses non-firm-specific news to generate ChatGPT signals, shows rather improved performance. Figure 3 graphically provides supportive evidence that Panel A (using all news) and Panel C (excluding *Firm-Specific News*) achieve higher cumulative returns than Panel B (only *Firm-Specific News*).

In summary, these results support the idea that ChatGPT can generate stock tickers from seemingly non-firm-specific news to optimize investment outcomes.

Electronic copy available at: https://ssrn.com/abstract=4759311

4.4. Mechanism and Interpretability

This section examines the mechanism of generating stock tickers from news articles that are mostly non-firm-specific news and the interpretability of the outcome from ChatGPT.

4.4.1. Mechanism

As the mechanism, I present the most important keywords for ChatGPT to generate its outcomes. The most direct way to figure out the reason for ChatGPT selecting certain stock tickers would be to ask again about the reason.

Table 8 directly implements the idea. This table reports topics for the reason for selecting the buy and sell lists that ChatGPT provides. The reasons are directly obtained from ChatGPT. The procedure is as follows. For each batch and random seed for 30 iterations, after obtaining the lists of stock tickers to buy and sell from ChatGPT, I ask ChatGPT again for the reason for selecting those stocks for each buy and sell list. Through this process, I can obtain 19,800 reasons for selecting each of the buy and sell lists. Several examples look like the following.

> ***BUY* Reason**
>
> The buy list comprises technology stocks like AAPL, GOOGL, AMZN, MSFT, and others, as the news indicates a positive outlook for technological innovation and growth. Companies like TSLA, NVDA, and META are included due to their potential in emerging technologies and market trends. Consumer discretionary stocks like NFLX, DIS, and COST have been chosen based on consumer spending trends and market sentiments. Biotech and pharmaceutical companies like MRNA are included due to potential advancements in healthcare. Cryptocurrency-related stocks like MARA and RIOT have been included considering market trends in digital assets.

Electronic copy available at: https://ssrn.com/abstract=4759311

SELL Reason

> The sell list includes bank stocks like UBS, GE, and BAC, as banking industry downturns are indicated. Stocks like FEDEX and WFC are included due to potential challenges in their respective sectors. Retail and real estate stocks like SPG and OUT are part of the sell list based on market projections. Cybersecurity stocks like PFPT and technology stocks such as MS and SBNY are included due to market performance trends. Energy companies like XOM have been chosen based on their sector performance. The diversified sell list covers various sectors with potential challenges identified in the news articles analyzed.

The reason for the *BUY* and *SELL* lists suggests various ideas, such as sector- or industry-level information and market sentiments.

For a systematic analysis, I use the Latent Dirichlet Allocation (LDA) model to extract the most important topics for each buy and sell reason.[14] The column *Topic for BUY Reason* reports the top 20 LDA-based keywords selected by their probability in the topic of the buy reasons. Similarly, *Topic for SELL Reason* reports the top 20 LDA-based keywords for the topic of the sell reasons.[15]

The results show that keywords in *Topic for BUY Reason* might be related to specific companies in tech sectors. However, more importantly, as intraday performance is attributable to the short leg of the strategy, keywords in *Topic for SELL Reason* are worth examining. Keywords with higher importance include “sector,” “industry,” “regulatory,” “decline,” “uncertainty,” “risk,” and several other specific keywords indicating sector names, such as “energy.” This result implies that the *SELL* side of the strategy generates output by analyzing input news articles to extract sector- or industry-level information possessing some negative sentiments and linking them to specific firms in some relevant industry mentioned in the news articles.

---

[14] The LDA model is used in empirical asset pricing literature to process news data (e.g., Bybee, Kelly, and Su (2023)).

[15] All keywords are lemmatized and lowercase by the Python Natural Language Toolkit (NLTK).

Electronic copy available at: https://ssrn.com/abstract=4759311

4.4.2. Economic Link

Table 9 reports the analysis results of examining the economic link between the generated stock tickers. The first consideration is the industry-level concentration of ChatGPT signals. The second consideration is the supply chain linkage between selected stocks.

To conduct portfolio-level analysis for each buy and sell list, the Propensity Score Matching (PSM) is used to match stocks with *BUY* and *SELL* signals with stocks that are not selected by ChatGPT neither as buy nor sell list.

Panel A compares the industry-level concentration between the stocks in the ChatGPT-generated lists and their matched lists of stocks. Columns (1) to (4) report the results for the buy list consisting of stocks having the *NET_BUY* signal. Industry-level concentration is measured by the Herfindahl-Hirschman Index (HHI). HHI is calculated for each day.

Column (1) uses SIC1, which is the 1-digit SIC code, to calculate HHI for the buy list and its matching group. The main dependent variable is the difference between the HHI of the buy list and the HHI of the matching group. The time-series difference of HHI is regressed on the constant term to estimate the time-series average of the difference.

The estimated difference between them is -0.016, which is highly significant.[16] This indicates that the ChatGPT-generated buy list has a lower industry-level concentration. This result implies that ChatGPT constructs a list of stocks that are diversified across various industries. A similar pattern is observed using different levels of industry definition in Columns (2), (3), and (4). In the sell list, the ChatGPT portfolio represents more diversified industries than the matching group. In Column (5), the ChatGPT sell list has 0.034 lower HHI than the matching group. Columns (6) to (8) are consistent that using different levels of industry definition provides similar results that the ChatGPT portfolio is more diverse than the matching group.

This result is counter-intuitive in that the previous evidence indicates that ChatGPT's stock selection might have some roles in sector- or industry-level analysis takes place. This might be attributable to the fact that ChatGPT can also consider other sources of economic linkages, such as supply chains, which

---

[16] The highly significant estimation results are attributable to the persistence of ChatGPT signal, which will be discussed in Section 4.7.

Electronic copy available at: https://ssrn.com/abstract=4759311

are not necessarily in the same industry. Furthermore, supply chains are a well-documented economic link between firms (e.g., Cohen and Frazzini (2008)).

Panel B compares the proportion of stocks that are linked as supply chains for the ChatGPT-generated lists and their matched lists of stocks. I calculated the proportion of stocks that are linked through the supply chain in each of the buy list and its matching group.

Column (1) uses supply chain data ranging from 2010 to 2022 (2010s) to get the supply chain linkage. The result shows that firms in the *NET_BUY* list have a 12.8% higher proportion of stocks that are linked as supply chains than its matching group. As a more illustration, the time-series average of the proportion of stocks having supply chain linkage in the *NET_BUY* list is 14.67%, and that of the matching group is 1.87%. The estimation result suggests that the difference is highly statistically significant.

Interestingly, when I use older data to calculate the supply chain linkage, the proportion of firms linked by the supply chain decreases. Columns (2) to (4) shows the results. For example, in Column (4), as supply chain data of the 1980s is used, stocks in the *NET_BUY* lists show a 1.4% higher probability of having supply chain-linked firms in the list than their matching group stocks. It is substantially lower compared to Column (1). This implies that the training data for ChatGPT might be tilted toward the recent sample period.

The results using the *NET_SELL* list in Columns (5) to (8) are also consistent. Column (5) shows that firms in *NET_SELL* lists have a 5.2% higher proportion of stocks that are linked as supply chains than its matching group. As a more illustration, the time-series average of the proportion of stocks having supply chain linkage in the *NET_SELL* list is 6.68%, and that of the matching group is 1.44%. For the *NET_SELL* list, when more past information is used, then the decay of identification for supply chain linkage is faster than that of the *NET_BUY* list. For example, when the supply chain data of the 1990s is used, then stocks in the *NET_SELL* list do not show the difference in the proportion of supply chain linkage to that of the matching group.

In summary, the results imply that lower industry-level concentration can be attributable to selecting stock in supply chain-linked firms.

Electronic copy available at: https://ssrn.com/abstract=4759311

4.4.3. Economic Link and Portfolio Performance

Table 10 reports various portfolio performances to examine the economic mechanism of ChatGPT's signal generation. Previous results suggest that ChatGPT considers "industry" or "sector" as important reasons for selecting the *NET_SELL* list, while their industry concentration is lower than their matching group. Then, the pattern might be attributable to ChatGPT's ability to select particular stocks that are most relevant to ChatGPT's industry-level analysis results. Therefore, a testable hypothesis is whether the performance comes from within-industry stock selection or across-industry stock selection.

Panel A reports the results for portfolios constructed by subsamples of stocks classified by within-industry signal. For each day and each SIC 2-digit code, a stock is classified as the *High* group if its *NET_BUY* or *NET_SELL* signal is above the median of the group and as the *Low* group when the signals are below the median. On the other hand, Panel B reports the results for portfolios constructed by subsamples of stocks classified by the across-industry signal. For each day, the across-industry signal is obtained by constructing SIC 2-digit-level portfolios and calculating the median of the *NET_BUY* and *NET_SELL* signals of the industry portfolios. Then, a stock is classified as the *High* group when the industry that includes the stock has the above the median *NET_BUY* or *NET_SELL* signal and as the *Low* group when below the median.

The results show that within-industry stock selection provides better long-short returns. Panel A shows that in the *High* group classified by the within-industry signal, *NET_BMS* shows a 4.745% of significant monthly CAPM alpha. On the other hand, Panel B shows that the *High* group classified by the across-industry signal exhibits relatively lower CAPM alpha as 4.062% per month.

Panel C reports the results of portfolios constructed by subsamples of stocks linked through the supply chain links or not. For each day and each of the stocks with the *NET_BUY* signal, stocks are classified as the *Yes* group when they have supply chain customers among other stocks with the *NET_BUY* signal and as the *No* group otherwise. Stocks with the *NET_SELL* signal are similarly categorized as the *Yes* or *No* group using supply chain data. The *Yes* group with the *NET_BUY* (*NET_SELL*) signal has 65 (40) stocks per day on average, while the *No* group has 377 (562) stocks per day on average.

The results indicate that the *Yes* group, which uses stocks having a supply chain, shows *NET_BMS* returns as 2.288%, whereas in the *No* group, *NET_BMS* is 3.821%. However, the results should be interpreted with caution since the numbers of stocks in the groups are highly unbalanced.

Electronic copy available at: https://ssrn.com/abstract=4759311

4.5. High-Frequency Performance of Portfolios by ChatGPT Signals and Notable Patterns

This paper focuses on day trading. Therefore, this section reports high-frequency performance and notable patterns observed in the high-frequency performance.

Table 11 reports the 30-minute interval risk-adjusted returns of ChatGPT day trading signals using NYSE Trade and Quote (TAQ) quote midpoint returns.[17] From 9:30 AM to 10:00 AM, the *NET_BUY* portfolio earns 0.848 bps of insignificant risk-adjusted returns. On the other hand, the *NET_SELL* portfolio shows -5.678 bps of significantly negative risk-adjusted returns. Therefore, the *NET_BMS* strategy earns significantly positive risk-adjusted returns of 6.527 bps. From 10:00 AM to the market closing, the *NET_BUY* strategy shows significantly positive returns at 1.395 bps from 11:00 AM to 11:30 AM, while the *NET_SELL* strategy shows significantly negative returns from 11:30 AM to 12:00 PM and 3:30 PM to 4:00 PM. As a result of the significantly negative risk-adjusted returns of *NET_SELL*, the *NET_BMS* portfolio shows significantly positive returns. However, the overnight return of the *NET_SELL* portfolio is significantly positive, 7.954 bps.

Figure 4 shows the high-frequency performance of the ChatGPT strategy. Using the TAQ, I generate 5-minute interval returns using quote midpoint returns. *NET_BUY*, *NET_SELL*, and *NET_BMS* are portfolio returns by *NET_BUY* signal, *NET_SELL* signal, and their differences. Using 271 days of intraday returns, I calculate each 5-minute interval returns for each time. $Day\ t$ is the portfolio formation day, $Day\ t-1$ is the day before portfolio day, and $Day\ t+1$ is the day after portfolio formation day.

The figure shows that the *NET_BUY* strategy earns returns gradually within a day. However, the *NET_SELL* strategy earns negative returns until 11:00 AM and is quite flat until the market closes. Moreover, the *NET_SELL* strategy clearly shows a strong overnight reversal from $Day\ t$ to $Day\ t+1$. The *NET_SELL* stocks clearly experienced negative returns on $Day\ t-1$, the day for the input news to generate ChatGPT signals. On the other hand, the *NET_BUY* strategy earns a moderate level of positive returns on $Day\ t-1$.

Besides the remarkable intraday performance, two notable patterns are observed in the figure. First, the *NET_SELL* portfolio experiences substantial overnight reversal, which is clearly observed in both the last row of Table 11 and Figure 4. Although the intraday returns of the *NET_SELL* portfolio show nearly 10 basis points decrease until the market closing, the overnight reversal is even greater than the intraday performance. This pattern raises significant concerns about using ChatGPT signals for overnight risk

---

[17] Quote midpoint returns are robust from bid-ask bounce (e.g., Pelger (2020), Bogousslavsky (2021)).

Electronic copy available at: https://ssrn.com/abstract=4759311

management. Addressing the issue is crucial to understanding the characteristics of ChatGPT signals and their implication on asset management.

Second, the ChatGPT-based trading strategy shows significant intraday performance on $Day\ t+1$ observed in Figure 4. This is notable since the portfolio is constructed at the beginning of $Day\ t$ using news articles compiled during the market opening of $Day\ t-1$ to the market opening of $Day\ t$. As news articles are considered to possess short-lived information content (e.g., Ke, Kelly, and Xiu (2019)), the pattern is counter-intuitive.

Investigating the rationale for these patterns would be helpful for understanding the nature of ChatGPT signals. Starting from the next section, I address those issues: first, I investigate the overnight holding performance, and second, I examine the persistence of ChatGPT day trading signals.

4.6. Overnight Performance

I start by showing the overnight performance of portfolios constructed by using ChatGPT signals. Table 12 reports the overnight performance as CAPM alpha. First, I report for the *Open-to-Close* return, which holds stocks from the market close of day t to the market opening of the next trading day. The ChatGPT signals for this strategy are based on tweets posted during the previous 24 hours of day t's market opening. Portfolio returns are regressed on overnight market factors. I report the CAPM alpha of the estimation. Second, I report for the *Open-to-Open* returns, which is holding stocks from the market close of day t to the market opening of the next trading day.

Panel A shows that the *BUY* portfolio shows a significantly positive overnight CAPM alpha of 0.659% per month, whereas the *SELL* portfolio shows an even higher CAPM alpha of 1.109%. *BMS* is -0.450%. Although intraday trading is profitable, none of the *Open-to-Open* returns are profitable. In Panels B and C, portfolios by *SUM_BUY*, *SUM_SELL*, *NET_BUY*, and *NET_SELL* signals present similar patterns. The short leg exhibits much higher returns than the long lag, resulting in negative *SUM_BMS* or *NET_BMS* returns. Accordingly, *Open-to-Open* return is not significant. Figure 5 graphically shows the results.

I test three ideas for the overnight return reversal. First, I test retail investors' overnight order imbalance motivated by similar trading patterns among retail investors (e.g., Berkman, Koch, Tuttle, and Zhang (2012)). Second, I test whether more timely portfolio formation can prevent overnight reversal. Third,

Electronic copy available at: https://ssrn.com/abstract=4759311

I test whether the prompt to ask ChatGPT to pretend to be a *day trader* is attributable to the overnight reversal.

4.6.1. Retail Overnight Order Imbalance

Berkman et al. (2012) suggest that overnight reversal is attributable to retail investors' attention-driven excessive net buying pressure. Therefore, the *NET_SELL* portfolio's substantial overnight reversal can be related to the attention of retail investors.

The main dependent variable to test the idea is the *Overnight Order Imbalance*. Using TAQ data, order direction is measured by following Lee and Ready (1991). *Overnight Order Imbalance* on day t is the difference between buying and selling orders divided by the sum of buying and selling orders during the market closing at day t to the market opening of day $t + 1$. Retail investors' trading is measured by following Boehmer, Jones, Zhang, and Zhang (2021).

Table 13 reports the panel regression results of analyzing the relationship between the overnight order imbalance by retail investors and ChatGPT day trading signals. Column (1) uses all trades to calculate *Overnight Order Imbalance*. The coefficient on *NET_BUY* shows that one more time selection in the buy list statistically significantly induces a 0.4% decrease of 1 standard deviation of *Overnight Order Imbalance*.[18] The coefficient on *NET_SELL* is statistically insignificant. The F-test of testing whether the coefficients of *NET_BUY* and *NET_SELL* are the same is not rejected as the F-statistics is 1.213 with a p-value much greater than 0.1.

Column (2) uses retail trades to calculate *Overnight Order Imbalance*, which is the difference between retail buying and selling orders divided by the sum of retail buying and selling orders. Retail investors show a significant increase in net buying for stocks selected in both buy and sell lists. As a stock is selected one more time in the buy list, order imbalance by retail investors increases by 1.5% of its 1 standard deviation.[19] On the other hand, as a stock is selected one more time in the sell list, order imbalance by retail investors increases by 3.4% of its 1 standard deviation. Therefore, the result shows that stocks with *NET_SELL* signals tend to have more retail investors' buying pressure. The F-test of testing whether the

---

[18] The 1 standard deviation of *Overnight Order Imbalance* is 0.785, resulting in the calculation 0.004=0.003/0.785.

[19] The 1 standard deviation of retail investors' *Overnight Order Imbalance* is 0.785, resulting in the calculation 0.015=0.013/0.850.

Electronic copy available at: https://ssrn.com/abstract=4759311

coefficients of *NET_BUY* and *NET_SELL* are the same is rejected as the F-statistics is 26.962 with less than 1% statistical significance.

Column (3) uses non-retail trades to calculate *Overnight Order Imbalance*. Non-retail trades are trades that are not classified as retail orders. The result is qualitatively similar to using all orders, which shows that stocks with the *NET_BUY* signals tend to experience weak overnight net selling price pressures, whereas stocks with the *NET_SELL* do not show a significant change in order imbalance.

Columns (4), (5), and (6) repeat Columns (1), (2), and (3) by using a subsample consisting of stocks having non-zero *NET_BUY* or *NET_SELL* signals. The results are qualitatively similar, suggesting that stocks with the *NET_SELL* signal tend to experience net buying price pressures by retail investors.

In summary, the results suggest that stocks that appear more in the sell list by ChatGPT tend to attract retail investors' attention, which results in the overnight return reversal.

4.6.2. *Intraday News* and *Overnight News*

The overnight return reversal can be attributable to the timing of portfolio construction. I construct portfolios at the market opening time using the previous 24 hours of news articles. The overnight return reversal happens from the market closing on the same day of the portfolio formation to the next day's market opening. Therefore, considering the potentially short-lived nature of information contents in the news data, more timely portfolio formation can prevent overnight return reversal.

Table 14 shows the portfolio performance of ChatGPT signals constructed by using different sets of input news data. *Intraday News* indicates tweets posted from 9:30 AM to 4:00 PM. *Overnight News* indicates tweets posted from 4:00 PM of a day to the next trading day's market opening at 9:30 AM. I report portfolio performance for *NET_BUY* and *NET_SELL* signals, and their differences, *NET_BMS*. All CAPM alphas are reported in monthly percentage terms.

Panel A shows that when *Overnight Return* at day t by using *Intraday News* at day t shows negative *NET_BMS* returns as -1.526%, which is still statistically significant. As *Intraday News* is news collected from 9:30 AM to 4:00 PM of day t, and *Overnight Return* at day t starts from 4:00 PM, the *Intraday News* at t is the most recent news to make investment decisions for *Overnight Return* at day t. The result indicates that more timely portfolio formation does not lead to improved overnight performance.

Electronic copy available at: https://ssrn.com/abstract=4759311

Moreover, Panels B, C, and D show that regardless of the input news, ChatGPT signals produce profitable intraday trading and substantial overnight reversal. Panel B shows that the overnight CAPM alpha of the *NET_BMS* strategy is -1.465% when one-day lagged *Overnight News* is used. Panel C shows that when only *Overnight News*, the most recent tweet for *Intraday Return*s, is used, *NET_BMS* earns 3.122%. Panel D reports a significant CAPM alpha of 3.765% for *Intraday Return* at day t using ChatGPT signals based only on *Intraday News* at day $t-1$. Figure 6 shows a clear pattern that regardless of the timing of the news, intraday trading is profitable, but it is followed by a substantial overnight reversal.

Table 15 further shows contemporaneous portfolio formation using overnight news to examine overnight performance. The results show that negative returns in *NET_BMS* disappear once I account for the contemporaneous information to construct portfolios, which is infeasible in real-time trading. In the usual return predictability analysis, including contemporaneous information tends to make spuriously outstanding performance. However, although I deliberately include future information at the moment of the portfolio formation, ChatGPT fails to produce significant returns in an overnight period. The results indicate that ChatGPT signals are not successful in dealing with overnight return patterns.

In summary, timely portfolio formation does not improve overnight performance even after accounting for contemporaneous information. Furthermore, more lagged news does not deteriorate the profitability of the day trading strategy. *Intraday News* and *Overnight News* are systematically not different for ChatGPT to generate signals.

4.6.3. Day Trader, Swing Trader, and Trader

My prompts ask ChatGPT to generate stock tickers for *day trading.*[20] Failure of ChatGPT to predict overnight returns can be attributable to this prompt engineering. In this subsection, I report the results by providing different trading styles to ChatGPT.

Swing Trader refers to traders who hold stocks for several days to several weeks.[21] Table 16 reports the portfolio performance of ChatGPT day trading signals generated using different prompts. All portfolio

---

[20] ChatGPT clearly understands the trading style of day trader. If I ask ChatGPT to pretend as a day trader and ask about trading horizon, then ChatGPT answers like the following: “As a day trader, my trading horizon typically spans a single trading day. This means that I focus on making short-term trades, entering and exiting positions within the same trading session to capitalize on intraday price movements.”

[21] https://www.investopedia.com/terms/s/swingtrading.asp

Electronic copy available at: https://ssrn.com/abstract=4759311

returns are CAPM alpha reported in monthly percentage terms. Panel A shows the benchmark performance of asking ChatGPT to pretend to be a *Day Trader*. Panel B shows the performance of asking to be a *Swing Trader*. Panel B shows that when I ask ChatGPT to be a *Swing Trader*, it shows better return in *Open-to-Close*, not in overnight periods (*Open-to-Open*). It results in positive and significant *Open-to-Open* returns, but it is not attributable to *Close-to-Open* returns. Panel C shows the performance of asking to be a *Trader*, which is asking without any specific trading style. In this case, *NET_BMS* shows worse returns than the case of specifying a particular trading style. Figure 7 shows the results graphically. The results suggest that *Swing Trader* shows better performance for a longer period of the trading horizon, but it is not attributable to overnight returns. It poses a challenge for the ChatGPT-based method to deal with overnight returns.

In summary, the substantial overnight return reversal in the *NET_SELL* portfolio is attributable to retail investors' attention-driven net buying pressure. On the other hand, the lag between news articles and portfolio formation and ChatGPT prompt engineering does not provide convincing evidence for overnight reversal.

4.7. Persistent News and ChatGPT Day Trading Signals

This section discusses the persistence of ChatGPT signals, which is one of the two notable patterns observed in Figure 4. I start by presenting the persistent portfolio performance using the ChatGPT signals. Then, I test the persistence of news articles by extracting important keywords to show that they persistently appear in news articles. Since news articles are the input variables for ChatGPT, this might be the source persistence of ChatGPT signals and resulting portfolio performance.

4.7.1. Decay of Signals

The ChatGPT signals maintain their profitability for several days. Table 17 shows that lagged ChatGPT signals still significantly predict future returns. For example, when we lag one day for the portfolio formation and the generation of ChatGPT signal, *NET_BMS* return still earns 3.402% per month. However, it is also followed by a significant overnight reversal, which is -1.674%. With more lags ranging from 2 to 5, the ChatGPT strategy shows a consistent pattern of profitable intraday trading and subsequent strong reversal overnight.

Electronic copy available at: https://ssrn.com/abstract=4759311

4.7.2. Persistent News

In this subsection, I examine the persistence of the news data to explain the persistence of ChatGPT signals. Table 18 tests the persistence of news releases. My empirical strategy is to specify topics of news that are statistically and intuitively related to the ChatGPT-based day trading performance and examine how persistent some sets of news are within near days. Therefore, the first step is to figure out topics, or more precisely, keywords in my Twitter data.

Keywords are selected from their Term Frequency-Inverse Document Frequency (TF-IDF) scores. There are several steps to obtain TF-IDF scores. I start by processing raw text data using the Python Natural Language Toolkit (NLTK). I classify each word in tweets using Part-of-Speech (POS) tagging. Through the POS tagging, I use only nouns, verbs, adjectives, and adverbs. I also lemmatize all words.

The next step is to construct documents. Therefore, I aggregate the processed tweets at a daily level, which results in 271 documents. Using these 271 documents, I calculate TF-IDF using N-grams ranging from 2 to 10 by imposing the minimum document frequency to 20 times. The final selected number of keywords is 11,555.

The last step is to select some important keywords based on their TF-IDF scores since the score represents the importance of each keyword. I calculate the time-series average of TF-IDF scores for each keyword and select the first 1,000 keywords having the highest time-series average TF-IDF score.

As a result, examples of keywords having the highest TF-IDF scores are “interest rate,” “central bank,” “new york,” “wall street,” and “prime minister.” Furthermore, “elon musk” and “donald trump” have the 8th and 9th highest TF-IDF, respectively.[22]

Using these keywords, I estimate the persistence of keywords by estimating the auto-regressive coefficient of each keyword’s TF-IDF score. I run the following regression:

$$Keyword_t = Constant + AR \cdot Keyword_{t-1} + \epsilon_t.$$

---

[22] I report keywords in lower case, which is the raw from NLTK.

Electronic copy available at: https://ssrn.com/abstract=4759311

$Keyword_t$ is the TF-IDF value of each keyword at day t. Among 1,000 keywords, the average value of *AR Coef* is 0.271, and 80.7% of them have p-values less than 10%. This result suggests that important keywords in my tweet data are very persistent.

Panel A reports the results for the top 30 TF-IDF score keywords. The aforementioned keywords, such as "interest rate," "central bank," "new york," "wall street," and "prime minister." Furthermore, "elon musk" and "donald trump" have significant autoregressive coefficients. I also test my keywords' explanatory power for the various ChatGPT-based portfolio returns.

$$Intraday\ Excess\ Returns_t = Constant + \beta_{MKTRF} \cdot MKTRF_t + \beta_{Keyword} \cdot H\{Keyword_{t-1}\} + \epsilon_t.$$

$H\{Keyword_{t-1}\}$ is a dummy variable having 1 when the keyword's TF-IDF value is greater than its median.[23] $Intraday\ Excess\ Returns_t$ can be one of *NET_BUY*, *NET_SELL*, and *NET_BMS*. Portfolio returns are included in the regression after being converted into monthly percentage terms. For brevity, I only report $\beta_{Keyword}$ for time-series regressions

Panel A shows the explanatory power of the top 30 important keywords for *NET_BMS* returns. However, none of the keywords significantly predict *NET_BMS* returns with positive coefficients.[24] Therefore, I separately report the top 30 keywords in terms of their significance in predicting *NET_BMS* returns. Panel B shows that keywords such as "energy price," "treasury secretary janet yellen," and "chinese president xi jinping" significantly predict *NET_BMS* returns. Interestingly, "10 year" might indicate the yield curve difference between 10 and 2-year US treasury, which has been discussed intensively in 2023 to discuss the looming risks of recessions.

Appendix Table 2 further reports the top 31 to 90 keywords ranked by their significance. For example, in Panel A, keywords related to "collapse silicon valley" provide some insights into the nature of ChatGPT-based trading strategy.

---

[23] Therefore, some keywords with less than half of positive values, $H\{Keyword_{t-1}\}$ becomes a dummy variable of indicating its appearance.

[24] I also test conventional macroeconomic indicators' impact on the ChatGPT-based trading strategy. I collect 13 categories of macroeconomic indicators' release days from the Bloomberg Economic Calendar (ECO). Appendix Table 3 shows that macroeconomic indicators have neither contemporaneous nor predictive effects on NET_BMS returns regardless of the various sizes of windows covering the release of macroeconomic indicators.

Electronic copy available at: https://ssrn.com/abstract=4759311

In summary, keywords in tweets by major business news providers like Bloomberg and the Wall Street Journal are very persistent. In the following subsection, I directly test the persistence of ChatGPT signals.

4.7.3. Persistent ChatGPT Signals

Keywords in tweets are persistent. As they are the main input variables for ChatGPT signals, the signals can also be persistent. Table 19, Table 20, and Table 21 test the persistence of ChatGPT signals. All estimations include date-fixed effects. The standard errors are two-way clustered by stock- and day-level.

Table 19 reports the panel regression results of examining the persistence of ChatGPT day trading signals. *BUY* (*SELL*) is a dummy variable having 1 when a stock is selected by ChatGPT to buy (sell). *BUY* $(t-1)$ is the previous day's *BUY* signal, and *SELL* $(t-1)$ is the previous day's *SELL* signal.

Columns (1) to (4) use *BUY* as the main dependent variable. The results show that stocks selected as *BUY* list tend to be selected as *BUY* list the next day. Columns (5) to (8) show a similar pattern that the *SELL* signal is also persistent. However, the previous *BUY* signal also predicts the *SELL* signal of the next day and vice versa.

Table 20 tests the persistence of ChatGPT signals using *NET_BUY* and *NET_SELL* signals.[25] Column (1) shows that the previous *NET_BUY* signal is a much stronger determinant than the *NET_SELL* signal for the next day's *NET_BUY* signal, and Column (2) shows that the *NET_SELL* signal becomes insignificant after controlling stock characteristics. Column (3) also shows that *NET_SELL* is much stronger for the next day's *NET_SELL* signal. Column (4) shows that as a stock has a higher *NET_BUY*, then it tends to have less *NET_SELL* signal, while the persistence of the *NET_SELL* signal is still maintained after controlling stock characteristics.

Table 21 shows the persistence in the shorter horizon. The results show that adjacent overnight and intraday *NET_BUY* and *NET_SELL* signals are persistent. In summary, as the input news for ChatGPT is persistent, the output ChatGPT signals also exhibit persistence.

---

[25] The regression model includes a dummy variable, *D_NET_SELL*, which is 1 when a stock has a *NET_SELL* signal, to adjust the difference of means between stocks with *NET_BUY* and *NET_SELL* signals.

Electronic copy available at: https://ssrn.com/abstract=4759311

Through the analysis, I suggest that input news articles and resulting output ChatGPT signals are also persistent. This persistence becomes the rationale for the persistent performance of the ChatGPT-based day trading strategy.

4.8. Limits-to-arbitrage

Table 22 reports the portfolio performance conditional on limits-of-arbitrage. All portfolios are constructed based on *NET_BUY* and *NET_SELL* signals. I use two categories of limits-to-arbitrage. Panel A uses firm size as a proxy of limits-of-arbitrage. Firm size is measured by the previous trading day's market capitalization. In each of the stocks with positive *NET_BUY* or *NET_SELL* signals, I divide stocks into *Small* and *Large* firms based on their size.

Panel A shows that in small stocks, *NET_BMS* earns 4.285% per month after adjusting the market factor, i.e., CAPM alpha, whereas in large stocks, *NET_BMS* earns 2.236%. Moreover, *NET_SELL* shows substantial negative returns in small stocks.

I also consider the average of rank-transformed variables proxying trading frictions, including share turnover, illiquidity, bid-ask spread, and return volatility. In each of the stocks with positive *NET_BUY* or *NET_SELL* signals, I divide stocks into *High* and *Low* groups based on their average rank of trading frictions.

Panel B also shows similar results that in the high trading friction group, *NET_BMS* is much higher than that of the low friction group. Furthermore, the significant returns come from the short leg. These results imply that ChatGPT signals are related to mispricing.

4.9. Fama-Macbeth Regression

Appendix Table 4 reports the Fama-Macbeth regression results of regressing intraday excess returns on ChatGPT day trading signals. The main dependent variable is intraday excess return, which is open-to-close price returns in basis point terms. The main independent variables are *NET_BUY* and *NET_SELL* signals. I include various control variables, including market capitalization (*SIZE*), share turnover (*TURN*), and the indicator of firm-specific information (*D_TWTR*).

Electronic copy available at: https://ssrn.com/abstract=4759311

Columns (1) to (4) report the results of using the full sample. Before testing the return predictability of *NET_BUY* and *NET_SELL* signals separately, Column (1) uses *NET_SIG*, which is the difference between *SUM_BUY* and *SUM_SELL* signals. For example, if a stock has *SUM_BUY* signal as 10 and *SUM_SELL* signal as 7, then the *NET_SIG* is 3, and if a stock has *SUM_BUY* as 7 and *SUM_SELL* as 10, then *NET_SIG* is -3. Using *NET_SIG* as the main independent variable, Column (1) shows that one more increase in *NET_SIG* predicts statistically significant 1.198 bps of returns in the intraday trading session. However, Column (2) shows that the effect of *NET_SIG* is subsumed by the inclusion control variables.

Columns (3) and (4) use a regression model to separately estimate the return predictability of NET_BUY and NET_SELL signals. Without control variables in Column (3), the NET_BUY signal is a significant predictor of returns. However, with control variables in Column (4), the NET_SELL signal becomes the significant predictor for future returns. Columns (5) to (8) repeat the analysis using a subsample of stocks having non-zero NET_BUY or NET_SELL signals. Results are qualitatively similar. Appendix Table 5 further includes more stock-level control variables. The results are similar in that the NET_SELL signal is significant after controlling for various stock-level characteristics.

In summary, the results indicate that the NET_BUY signal mostly reflects stock-level characteristics, such as firm size or turnover. In contrast, the NET_SELL signal has more incremental information contents than stock-level characteristics.

4.10. Role of Prompts

4.10.1. Order in *BUY* and *SELL* Lists

In the prompts, I ask ChatGPT to generate the *BUY* and *SELL* lists in the order of most likely to go up comes first, and for the SELL list, vice versa. The reason for providing this prompt is to maximize the benefit of generating the role of ChatGPT. LLM-based Generative AIs, including ChatGPT, generate outputs based on tokens that have been generated. Therefore, including some direction for the output can be helpful for obtaining better output.

Appendix Table 6 and Appendix Table 7 test this idea. Appendix Table 6 presents the result for not specifying the order of lists. The results show that the CAPM alpha of *NET_BMS* in Panel C, 3.490%, is lower than my benchmark performance, 3.717%. On the other hand, Appendix Table 7 shows that the order-weighted lists are not necessarily good for ChatGPT to produce more profitable lists. The CAPM alpha of

Electronic copy available at: https://ssrn.com/abstract=4759311

NET_BMS in Panel C is 3.106%, which is lower than the benchmark case. In unreported analysis, Swing Trader shows substantially lower performance when I provide prompts without the direction of order. Therefore, the prompt engineering for asset pricing would be a fruitful area for further research.

5. Conclusion

This paper shows that ChatGPT can generate stock tickers to buy and sell for day trading. Unlike the prior literature, which provides only firm-specific news to ChatGPT to get firm-specific signals, I provide multiple firm-specific and non-firm-specific news together to generate lists of stock tickers to buy and sell. I show that a trading strategy based on ChatGPT's buy and sell signals earns significant long-short returns in open-to-close intraday trading.

Notably, most of the stock tickers generated by ChatGPT are not mentioned in Twitter news articles. It suggests that ChatGPT's generative feature works for empirical asset pricing to generate relevant outputs. Furthermore, the portfolio performance is not attributable to some stocks that are directly mentioned in the input news articles, which suggests that the *generative* feature of ChatGPT works to select relevant stocks by analyzing input news articles.

This paper also analyzes the reason for ChatGPT to generate certain stock tickers by asking again about the reasons for selecting those lists. The top keywords for selecting the buy lists represent tech stocks, whereas the top keywords for generating the sell lists are related to sector- or industry-specific considerations. Further analysis shows that the generative feature of ChatGPT selects stocks that are economically linked through supply chains, resulting in lower industry concentration in each of the buy and sell portfolios than their matching groups. The lower level of industry concentration is also consistent with additional results, suggesting ChatGPT's superior stock selection performance within each industry. The standard limits-to-arbitrage test suggests that the performance is mostly attributable to stocks with higher limits-to-arbitrage, implying ChatGPT's applicability of extracting mispricing signals from seemingly non-firm-specific news.

There are two notable patterns of the strategy. First, the sell side of the recommendation from ChatGPT brings significant retail attention. It induces substantial overnight return reversal for the short leg of the strategy. This finding poses a careful approach to using ChatGPT in empirical asset pricing due to their lack of predictability for overnight returns. Second, the ChatGPT-based trading signals are persistent. This is attributable to the persistence of the input news articles.

Electronic copy available at: https://ssrn.com/abstract=4759311

This paper sheds light on the literature by showing a new method of prompt engineering that provides multiple news articles, including both firm-specific and macro news. This can be considered as a factor model estimating latent factor structures from macroeconomic data or economy-wide variables, and estimating individual firms' exposure to the factors. My finding suggests the applicability of the LLM-based model to make a factor model.

Electronic copy available at: https://ssrn.com/abstract=4759311

References

Amihud, Y. 2002. Illiquidity and stock returns: cross-section and time-series effects. *Journal of financial markets* 5: 31-56.

Avramov, D., S. Cheng, and L. Metzker. 2023. Machine learning vs. economic restrictions: Evidence from stock return predictability. *Management Science* 69: 2587-619.

Bai, J. J., N. M. Boyson, Y. Cao, M. Liu, and C. Wan. 2023. Executives vs. chatbots: Unmasking insights through human-AI differences in earnings conference Q&A. In editor^editors. Nicole M. and Cao, Yi and Liu, Miao and Wan, Chi, Executives vs. Chatbots: Unmasking Insights through Human-AI Differences in Earnings Conference Q&A (June 15, 2023).

Beckmann, L., H. Beckmeyer, I. Filippou, S. Menze, and G. Zhou. 2024. Unusual Financial Communication-Evidence from ChatGPT, Earnings Calls, and the Stock Market. *Earnings Calls, and the Stock Market (January 15, 2024)*.

Berkman, H., P. D. Koch, L. Tuttle, and Y. J. Zhang. 2012. Paying attention: overnight returns and the hidden cost of buying at the open. *Journal of Financial and Quantitative Analysis* 47: 715-41.

Boehmer, E., C. M. Jones, X. Zhang, and X. Zhang. 2021. Tracking retail investor activity. *The Journal of Finance* 76: 2249-305.

Bogousslavsky, V. 2016. Infrequent rebalancing, return autocorrelation, and seasonality. *The Journal of Finance* 71: 2967-3006.

———. 2021. The cross-section of intraday and overnight returns. *Journal of Financial Economics* 141: 172-94.

Bogousslavsky, V., V. Fos, and D. Muravyev. 2022. Informed trading intensity. *The Journal of Finance*.

Bond, S. A., H. Klok, and M. Zhu. 2023. Large language models and financial market sentiment. *Available at SSRN 4584928*.

Bybee, L., B. Kelly, and Y. Su. 2023. Narrative asset pricing: Interpretable systematic risk factors from news text. *The Review of Financial Studies* 36: 4759-87.

Byun, S.-J., S. Cho, and D.-H. Kim. 2024. Can a machine learn from behavioral biases? Evidence from stock return predictability of deep learning models. *Journal of Behavioral and Experimental Finance* 41: 100881.

Chen, J., G. Tang, G. Zhou, and W. Zhu. 2023. ChatGPT, Stock Market Predictability and Links to the Macroeconomy. *Available at SSRN 4660148*.

Electronic copy available at: https://ssrn.com/abstract=4759311

Chen, L., M. Pelger, and J. Zhu. 2024. Deep learning in asset pricing. *Management Science* 70: 714-50.

Cohen, L., and A. Frazzini. 2008. Economic links and predictable returns. *The Journal of Finance* 63: 1977-2011.

Glasserman, P., and C. Lin. 2023. Assessing Look-Ahead Bias in Stock Return Predictions Generated By GPT Sentiment Analysis. *arXiv preprint arXiv:2309.17322*.

Gu, S., B. Kelly, and D. Xiu. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies* 33: 2223-73.

———. 2021. Autoencoder asset pricing models. *Journal of Econometrics* 222: 429-50.

Heston, S. L., R. A. Korajczyk, and R. Sadka. 2010. Intraday patterns in the cross-section of stock returns. *The Journal of Finance* 65: 1369-407.

Jha, M., J. Qian, M. Weber, and B. Yang. 2024. ChatGPT and corporate policies.

Ji, Z., N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys* 55: 1-38.

Karolyi, G. A., and S. Van Nieuwerburgh. 2020. New methods for the cross-section of returns. *The Review of Financial Studies* 33: 1879-90.

Ke, Z. T., B. T. Kelly, and D. Xiu. 2019. Predicting returns with text data.

Kelly, B. T., S. Pruitt, and Y. Su. 2019. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics* 134: 501-24.

Kim, A., M. Muhn, and V. Nikolaev. 2023a. From Transcripts to Insights: Uncovering Corporate Risks Using Generative AI. *arXiv preprint arXiv:2310.17721*.

Kim, A. G., M. Muhn, and V. V. Nikolaev. 2023b. Bloated disclosures: can ChatGPT help investors process information? *Chicago Booth Research Paper*.

Lee, C. M., and M. J. Ready. 1991. Inferring trade direction from intraday data. *The Journal of Finance* 46: 733-46.

Li, E. X., Z. Tu, and D. Zhou. 2023. The promise and peril of generative AI: Evidence from ChatGPT as sell-side analysts. *Available at SSRN 4480947*.

Li, K., F. Mai, R. Shen, C. Yang, and T. Zhang. 2023. Dissecting corporate culture using generative AI–Insights from analyst reports. *Available at SSRN 4558295*.

Lopez-Lira, A., and Y. Tang. 2023. Can chatgpt forecast stock price movements? return predictability and large language models. *arXiv preprint arXiv:2304.07619*.

Electronic copy available at: https://ssrn.com/abstract=4759311

Pelger, M. 2020. Understanding systematic risk: A high-frequency approach. *The Journal of Finance* 75: 2179-220.

Pelster, M., and J. Val. 2024. Can ChatGPT assist in picking stocks? *Finance Research Letters* 59: 104786.

Romanko, O., A. Narayan, and R. H. Kwon. 2023. Chatgpt-based investment portfolio selection. In editor^editors. Operations Research Forum, Vol. 4. Springer.

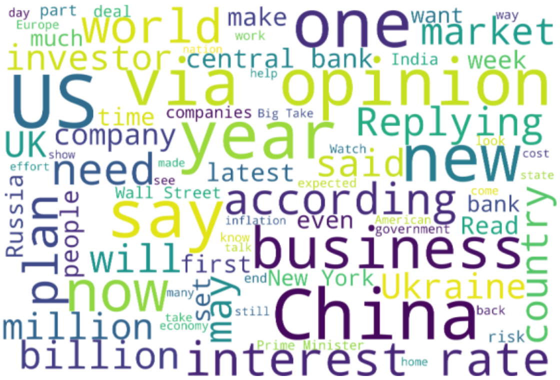
Vamossy, D. F., and R. Skog. 2023. Emtract: Extracting emotions from social media. *Available at SSRN 3975884*.

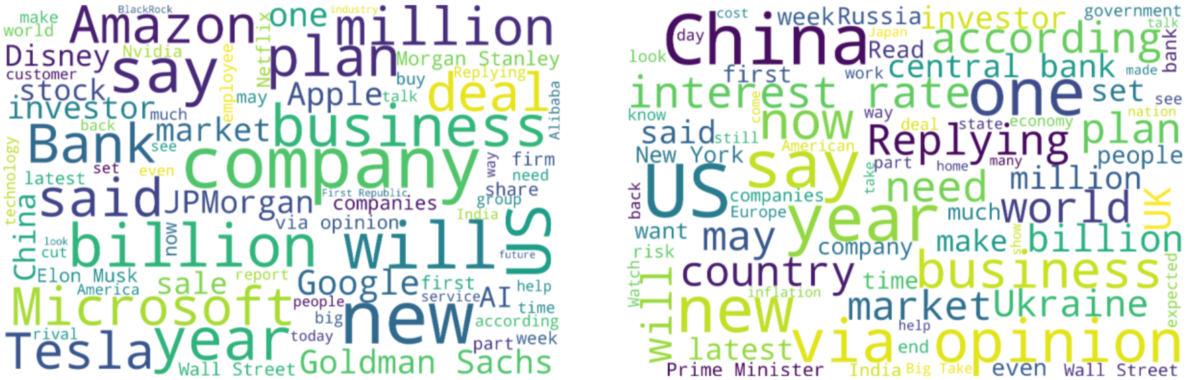
Electronic copy available at: https://ssrn.com/abstract=4759311

Figure 1. Word Cloud from News Articles posted on Twitter

This figure plots word clouds from tweets posted on Twitter (currently, X) by Bloomberg and the Wall Street Journal. Tweets are collected from Refinitiv Eikon News Monitor. If a tweet contains some firm-specific information, then Eikon provides relevant best-matching Refinitiv Instrument Code (RIC). *Firm-Specific News* indicates tweets having RIC information. The sample period is from December 1, 2022, to December 31, 2023. The total number of tweets used in this analysis is 222,659. The number of tweets with RIC information is 16,359. Panel A shows a word cloud using all news. Panel B shows a word cloud using only *Firm-Specific News*. Panel C shows a word cloud excluding *Firm-Specific News*.

Panel A. All News





Panel B. Only *Firm-Specific News*

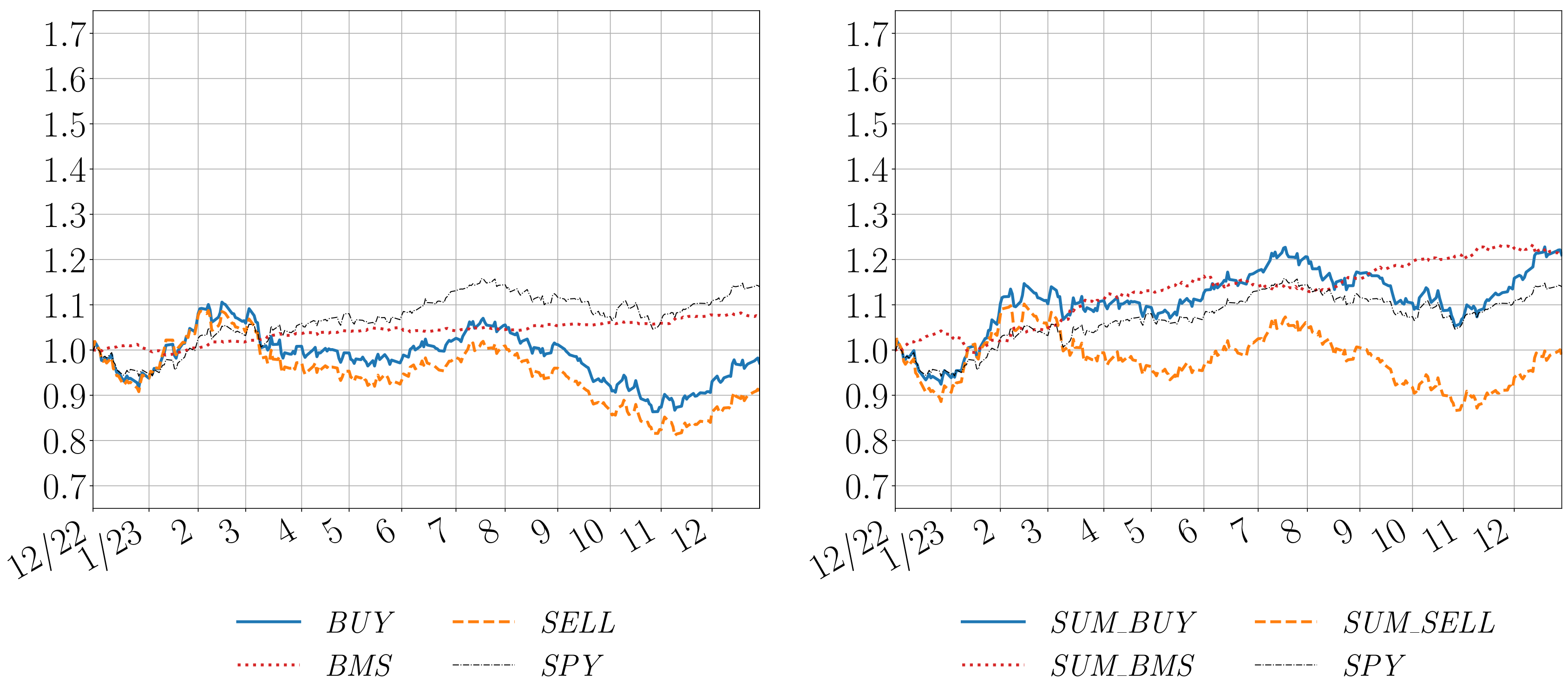
Panel C. Excluding *Firm-Specific News*

Electronic copy available at: https://ssrn.com/abstract=4759311

Figure 2. Intraday Portfolio Performance of ChatGPT Day Trading Signals

This figure plots the portfolio performance of the day trading strategy based on ChatGPT day trading signals. The sample period is from December 2022 to December 2023, which is 271 trading days. The *BUY* and *SELL* signals are constructed from ChatGPT by asking to pick stocks to buy and sell based on tweets posted on Twitter by Bloomberg and the Wall Street Journal. Specifically, for day t's day trading strategy, I ask ChatGPT to generate lists of stock tickers to buy and sell by providing tweets posted from the day $t-1$ market opening to the right before the market opening of day t at 9:30 AM US Eastern Standard Time (EST). A stock has a *BUY* (*SELL*) signal when it is picked by ChatGPT to buy (sell). Due to ChatGPT's limitation of the maximum number of tokens to process per request, I provide several batches of tweets to ensure the input tweets do not exceed the limit, which is 16,385 tokens, including both prompts and responses. For each batch of tweets, I repeat 30 times to ask to generate lists of stock tickers to buy and sell, which produces multiple lists of stocks to buy and sell. Panel A plots cumulative portfolio excess returns by the *BUY* and *SELL* signals and their difference, denoted by *BMS* (*BUY* minus *SELL*). The *BUY* (*SELL*) portfolio is constructed by holding stocks to buy (sell) that have at least one *BUY* (*SELL*) signal across the lists. We hold stocks from the market opening to closing. Accordingly, our portfolio return is open-to-close price returns. We proxy risk-free rates using open-to-close price returns of BIL (SPDR Bloomberg 1-3 Month T-Bill ETF). The excess return is the difference between portfolio returns and risk-free rates. Panel B plots cumulative portfolio excess returns constructed by using *SUM_BUY* and *SUM_SELL* signals and their long-short return, which is denoted by *SUM_BMS*. *SUM_BUY* (*SUM_SELL*) is the sum of *BUY* (*SELL*) signals across the lists. *SUM_BUY* (*SUM_SELL*) portfolio is constructed by weighting stocks using *SUM_BUY* (*SUM_SELL*) signals. Panel C plots cumulative portfolio excess returns constructed by using the *NET_BUY* and *NET_SELL* signal and their long-short return, which is denoted by *NET_BMS*. *NET_BUY* and *NET_SELL* signals are obtained by netting the *SUM_BUY* and *SUM_SELL* signals. *NET_BUY* (*NET_SELL*) portfolio is constructed by weighting stocks using *NET_BUY* (*NET_SELL*) signals.

Electronic copy available at: https://ssrn.com/abstract=4759311

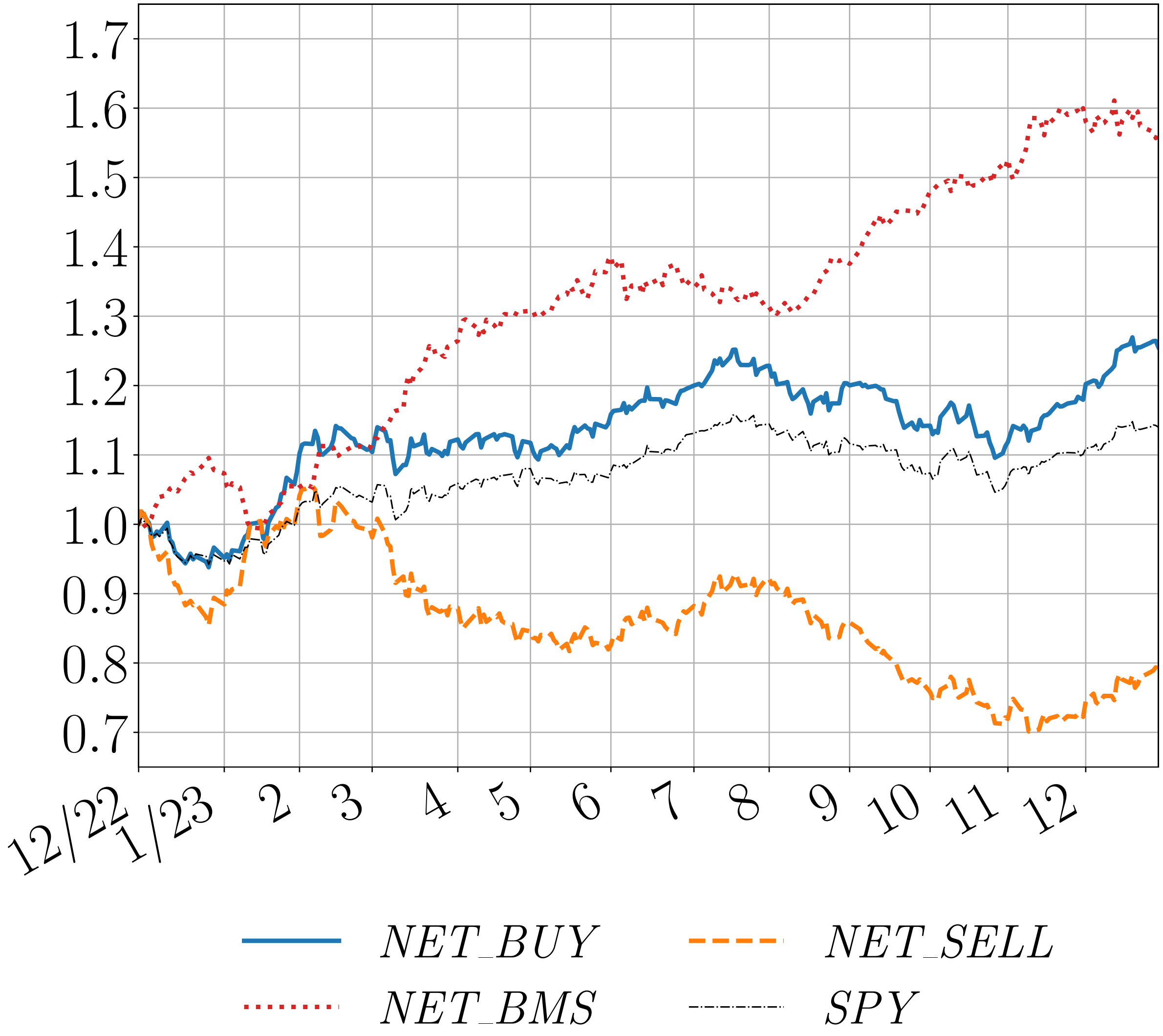


Panel A. *BUY* and *SELL* Signals

Panel B. *SUM_BUY* and *SUM_SELL* Signals

Electronic copy available at: https://ssrn.com/abstract=4759311

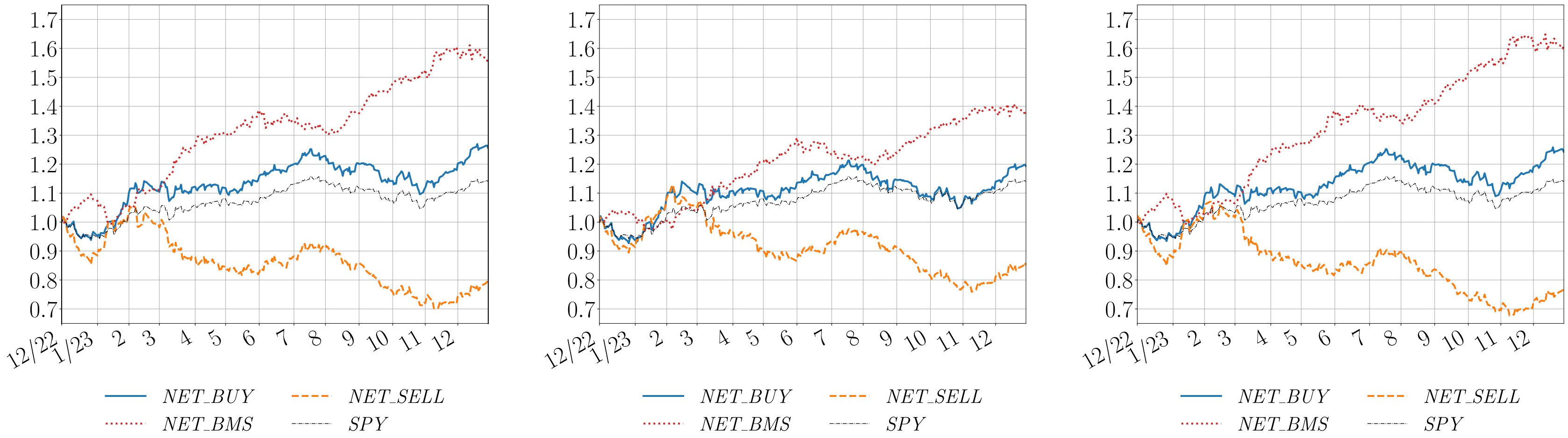
Panel C. *NET_BUY* and *NET_SELL* Signals



Electronic copy available at: https://ssrn.com/abstract=4759311

Figure 3. Performance of ChatGPT Day Trading Signal Using Only Firm-Specific News

This figure plots the performance of ChatGPT signals generated by using different sets of tweets. If a tweet contains some firm-specific information, then Eikon provides relevant best-matching Refinitiv Instrument Code (RIC). *Firm-Specific News* indicates tweets having RIC information. The total number of tweets used in this analysis is 222,659. The number of tweets with RIC information is 16,359. Panel A shows the benchmark performance by using all tweets (*All News*). Panel B shows the performance of using only *Firm-Specific News*. Panel C shows the performance of excluding *Firm-Specific News*). All portfolios are constructed by *NET_BUY* and *NET_SELL* signals.



Panel A. All News

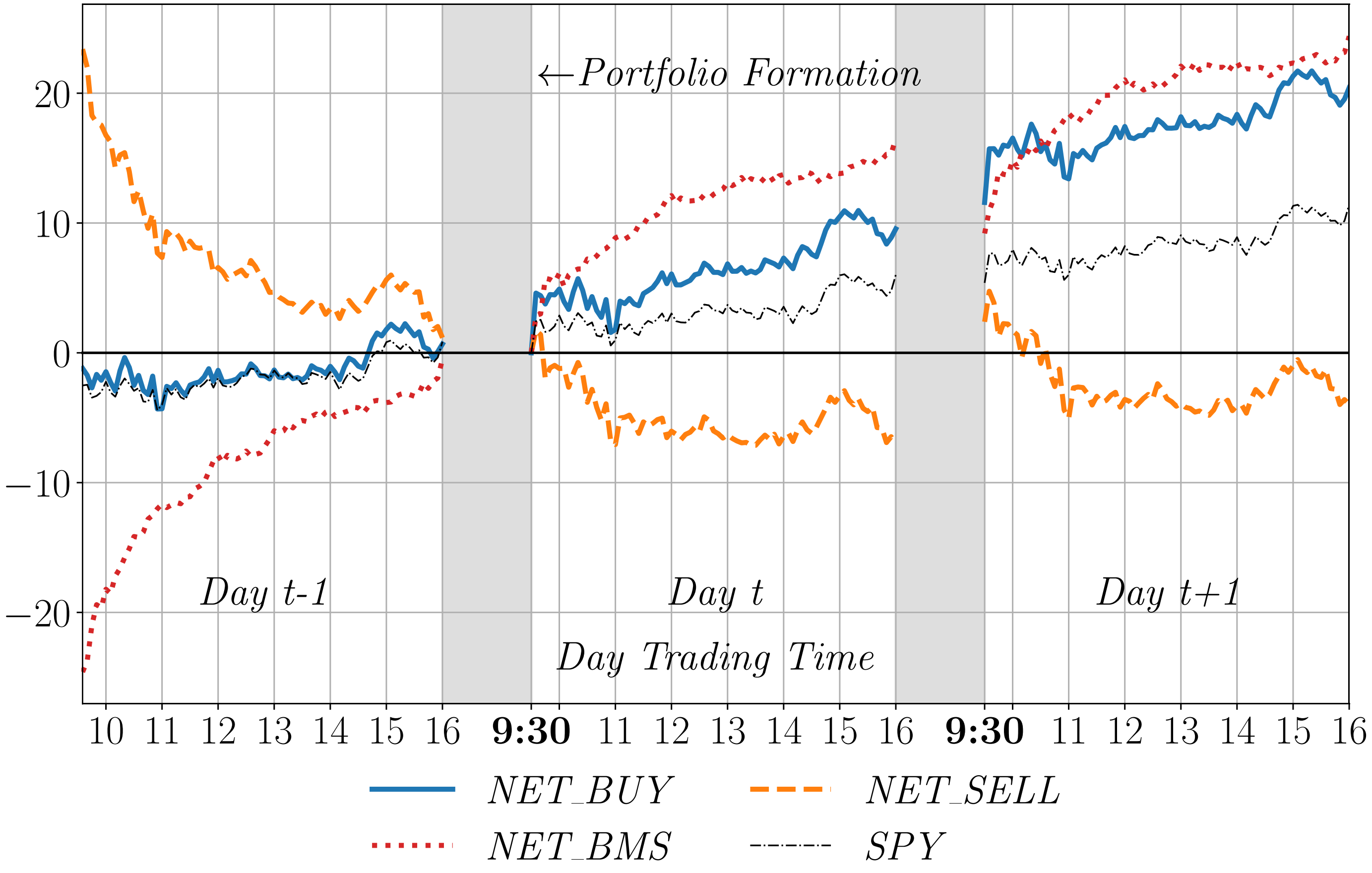
Panel B. Only *Firm-Specific News*

Panel C. Excluding *Firm-Specific News*

Electronic copy available at: https://ssrn.com/abstract=4759311

Figure 4. High-Frequency Performance of ChatGPT Day Trading Signal

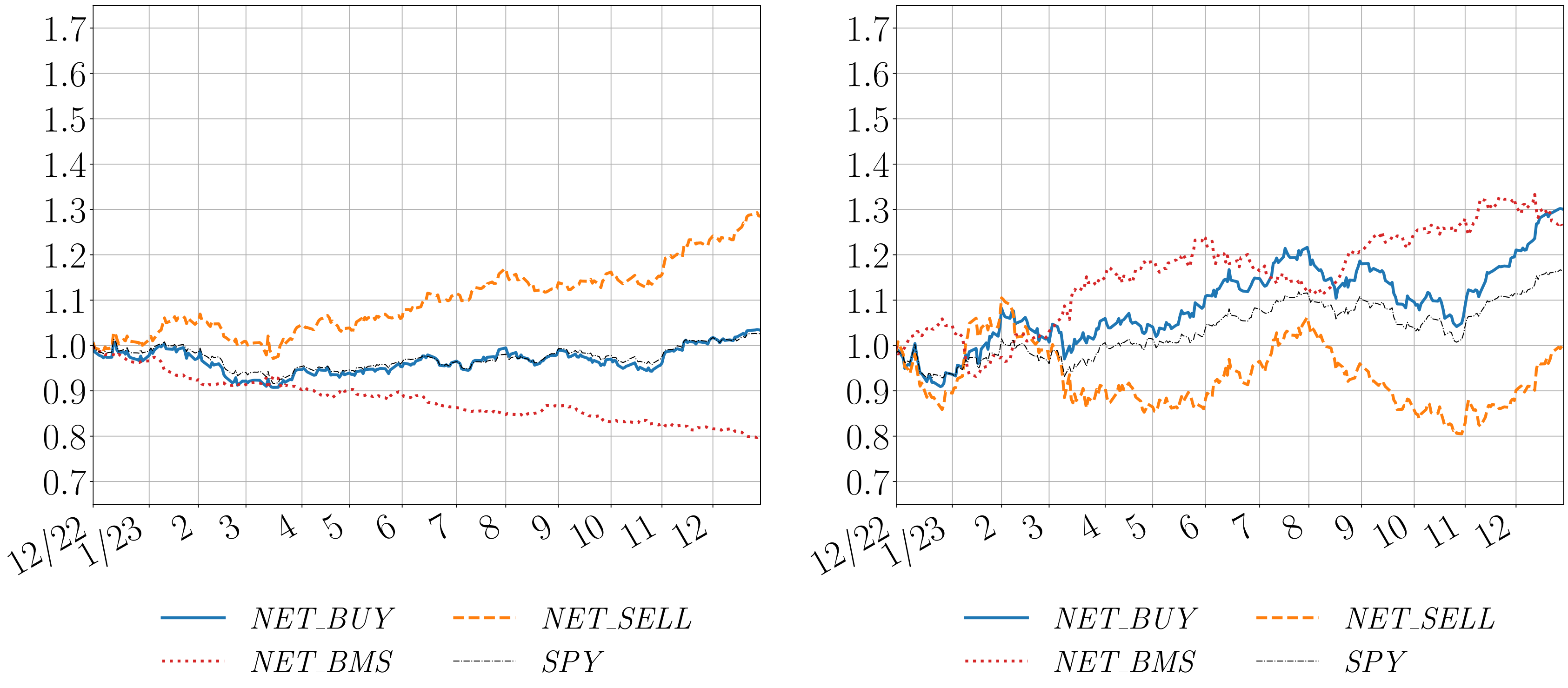
This figure plots the intraday-level performance of the ChatGPT day trading signal. Using the NYSE Trade and Quote (TAQ) database, I generate 5-minute interval returns using quote midpoint returns. *NET_BUY*, *NET_SELL*, and *NET_BMS* are portfolio returns by *NET_BUY* signal, *NET_SELL* signal, and their differences. The y-axis is basis points (bps). The shaded areas represent the overnight period, which is from 4:00 PM to the next day's market opening at 9:30 AM. Using 271 days of intraday returns, I calculate each 5-minute interval returns for each 5-minute interval. $Day\ t$ is the portfolio formation day, $Day\ t-1$ is the day before portfolio day, and $Day\ t+1$ is the day after portfolio formation day.



Electronic copy available at: https://ssrn.com/abstract=4759311

Figure 5. Overnight Performance of ChatGPT Day Trading Signal

This figure plots the overnight portfolio performance of *NET_BUY* and *NET_SELL* signals generated by ChatGPT. All signals at day t are generated by using the previous 24 hours of tweets, which is from 9:30 AM of day $t-1$ to right before 9:30 AM of day t. Panel A plots the cumulative portfolio performance of holding stocks from day t's closing to the next trading day's opening (*Close-to-Open*). Panel B plots the cumulative portfolio performance of holding stocks from day t's opening to the next trading day's opening (*Open-to-Open*). All portfolios are constructed by *NET_BUY* and *NET_SELL* signals.



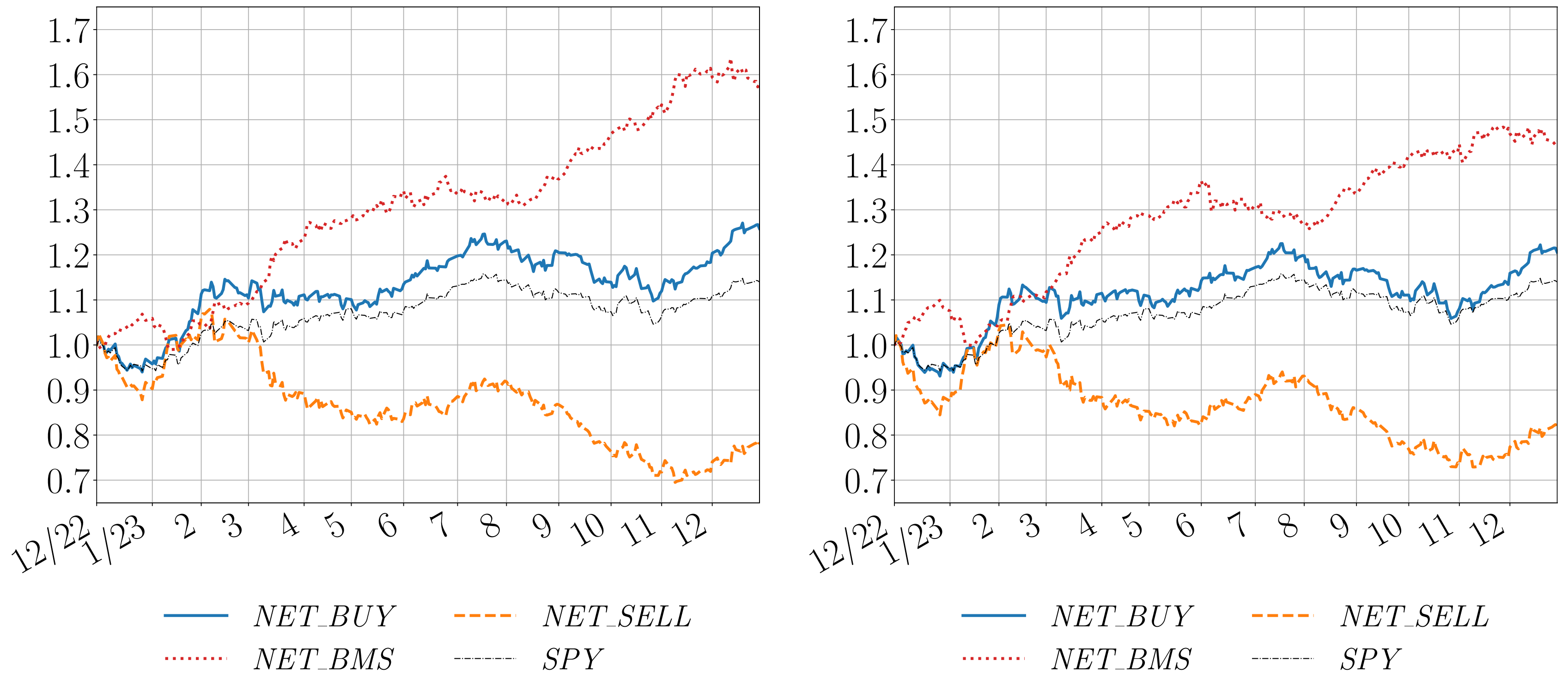
Panel A. *Close-to-Open*

Panel B. *Open-to-Open*

Electronic copy available at: https://ssrn.com/abstract=4759311

Figure 6. Performance of ChatGPT Day Trading Signals based on Intraday and *Overnight News*

This figure plots the portfolio performance of ChatGPT day trading signals generated by using different sets of tweets. *Intraday Return* indicates portfolio returns by holding stocks from 9:30 AM to 4:00 PM of day *t*. *Overnight Return* indicates portfolio returns by holding stocks from 4:00 PM of day *t* to the next trading day's market opening at 9:30 AM. *Intraday News* indicates tweets posted from 9:30 AM to 4:00 PM of day *t*. *Overnight News* indicates tweets posted from 4:00 PM of day *t* to the next trading day's market opening at 9:30 AM. All portfolios are constructed by appropriately lagged *NET_BUY* and *NET_SELL* signals. Panel A plots the cumulative *Intraday Return* using *Intraday News*. Panel B plots the cumulative *Intraday Return* using *Overnight News*. Panel C plots the cumulative *Overnight Return* using *Intraday News*. Panel D plots the cumulative *Overnight Return* using *Overnight News*.

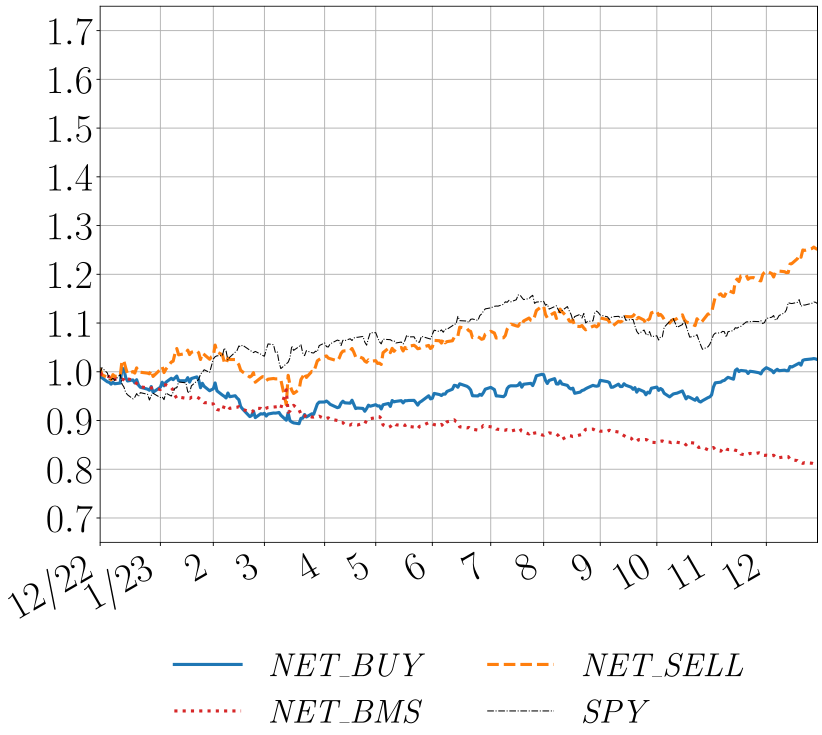


Panel A. *Intraday Return* by *Intraday News*

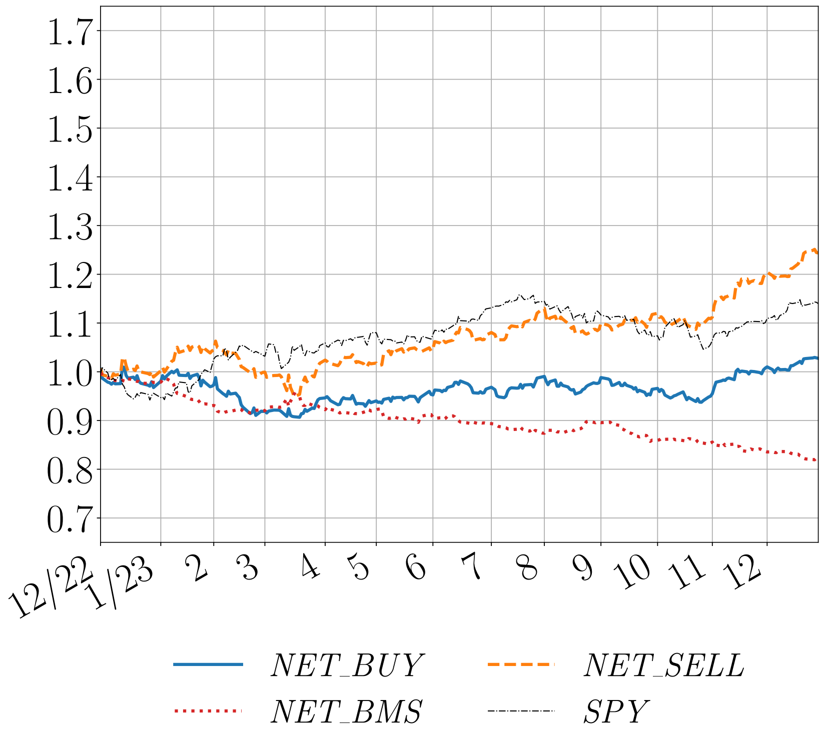
Panel B. *Intraday Return* by *Overnight News*

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel C. *Overnight Return* by *Intraday News*



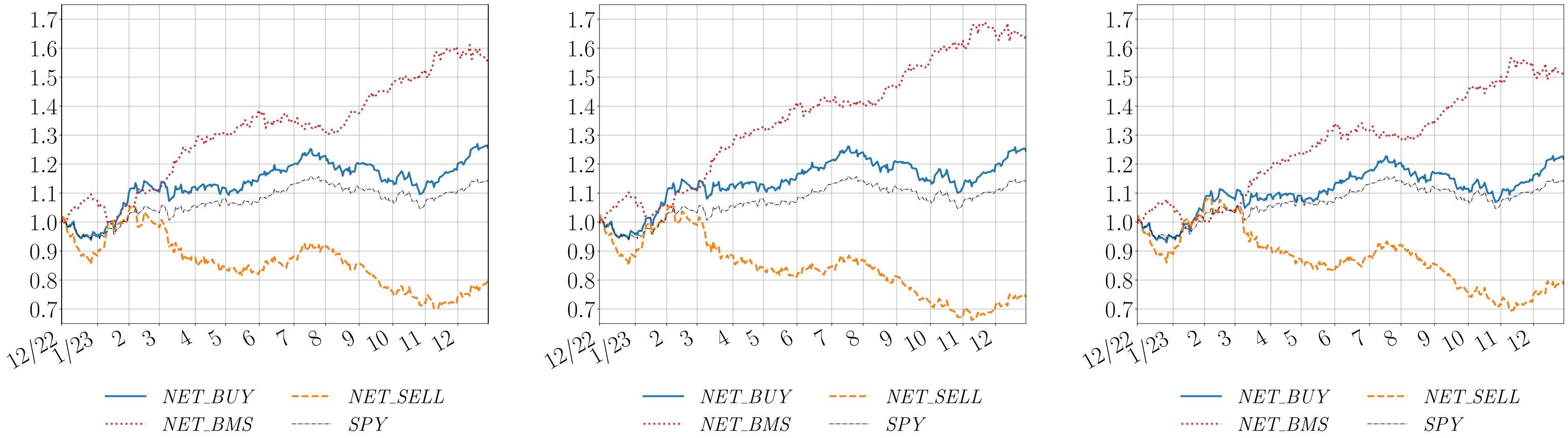
Panel D *Overnight Return* by *Overnight News*



Electronic copy available at: https://ssrn.com/abstract=4759311

Figure 7. Day Trader, Swing Trader, and Trader

This figure plots the portfolio performance of ChatGPT day trading signals generated by using different prompts. Panel A shows the benchmark performance of asking ChatGPT to pretend to be a *Day Trader*. Panel B shows the performance of asking to pretend to be a *Swing Trader*. Panel C shows the performance of asking to pretend to be a *Trader*, providing prompts without any specific trading style. All portfolios are constructed by *NET_BUY* and *NET_SELL* signals.



Panel A. *Day Trader*

Panel B. *Swing Trader*

Panel C. *Trader*

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 1. Examples of Tweets

This table presents examples of tweets posted on Twitter (currently, X) by Bloomberg. @business, @markets, and @opinion are Bloomberg, Bloomberg Markets, and Bloomberg Opinion, respectively. The posted time is based on the US Eastern Standard Time (EST). Each tweet can be seen through the provided Twitter link. Refinitiv Instrument Code (RIC) is provided by Refinitiv Eikon News Monitor. A tweet is classified as *Overnight News* when posted from the previous trading day's market close (4:00 PM) to the right before the next trading day's market opening (9:30 AM). A tweet is classified as *Intraday News* when posted during the market opening from 9:30 AM to 4:00 PM. *Overnight News* is used to predict the same day's intraday returns, whereas *Intraday News* is used to predict the next trading day's intraday returns.

Date	Time	Tweet	RIC	Intraday or Overnight?	Intraday to Predict
Jan 18, 2023 Wed	8:47:56 AM	@business: As job cuts loom, Microsoft CEO Satya Nadella says the technology industry must learn to be efficient as demand slows https://t.co/7UH5ro36Ri Twitter link: https://twitter.com/business/status/1615707566268375041	MSFT.O	*Overnight News*	Jan 18, 2023
Feb 7, 2023	9:34:42 AM	@markets: Bed Bath & Beyond plunges by a record 48% https://t.co/mUmPm5qCZX https://t.co/6cQcX4hVJ3 Twitter link: https://twitter.com/markets/status/1622967091887476736	BBBY.O	*Intraday News*	Feb 8, 2023
Nov 14, 2023, Tue	8:28:42 AM	@opinion: Replying to @opinion: CPI: The report is set to show slower progress toward the Federal Reserve's 2% inflation target, keeping the central		*Overnight News*	Nov 14, 2023

Electronic copy available at: https://ssrn.com/abstract=4759311

		bank biased toward more tightening. 🎥 @JonathanJLevin joins @GinaMartinAdams and @Claudia_Sahm to break down the latest inflation data https://t.co/zL119osH1W Twitter link: https://twitter.com/opinion/status/1724419086044516405		
Nov 14, 2023, Tue	8:32:39 AM	@opinion: Replying to @opinion: NEW: No rise in headline CPI, with core CPI rising just 0.2% (below estimate!). Watch live: @JonathanJLevin, @GinaMartinAdams and @Claudia_Sahm are breaking down the latest inflation data https://t.co/zL119osH1W Twitter link: https://twitter.com/opinion/status/1724420078274244771	*Overnight News*	Nov 14, 2023
Nov 14, 2023, Tue	8:37:06 AM	@opinion: Replying to @opinion: US CPI: ⚡ The energy index fell 2.5% ⛽ The gasoline index was down by 5.0% Watch live analysis ⬇ https://t.co/zL119osH1W Twitter link: https://twitter.com/opinion/status/1724421197901750727	*Overnight News*	Nov 14, 2023

Electronic copy available at: https://ssrn.com/abstract=4759311

Aug 1, 2023, Tue	2:40:09 PM	@markets: US job openings fell in June to the lowest level since April 2021, suggesting some softening in labor-market conditions https://t.co/wLD5WDsnee Twitter link: https://twitter.com/markets/status/1686446735785172993	*Intraday News*	Aug 2, 2023

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 2. Summary Statistics

This table reports summary statistics of ChatGPT day trading signals. The sample period is from December 2022 to December 2023, which is 271 trading days. The *BUY* and *SELL* signals are constructed from ChatGPT by asking to pick stocks to buy and sell based on tweets posted on Twitter by Bloomberg and the Wall Street Journal. Specifically, for day t's intraday trading strategy, I ask ChatGPT to generate lists of stock tickers to buy and sell by providing tweets posted from the day $t-1$ market opening to the right before the market opening of day t at 9:30 AM US Eastern Standard Time (EST). Due to ChatGPT's limitation of the maximum number of tokens to process per request, I provide several batches of tweets to ensure the input tweets do not exceed the limit, which is 16,385 tokens, including both prompts and responses. For each batch of tweets, I repeat 30 times to ask to generate lists of stock tickers to buy and sell, which produces multiple lists of stocks to buy and sell. Using stocks with CRSP share codes 10 and 11 that are traded in NYSE, AMEX, and NASDAQ, I construct a daily panel of US-listed equity samples. For each stock-day observation, the *BUY* (*SELL*) signal is a dummy variable having 1 when the stock's ticker appears at least once in the buy (sell) lists. If a tweet contains some firm-specific information, then Eikon provides relevant best-matching Refinitiv Instrument Code (RIC). *D_TWTR* is a dummy variable having 1 when a tweet has RIC information. Panel A reports the full sample summary statistics of *BUY* and *SELL* signals and *D_TWTR*. Panel B reports a detailed distribution of *SUM_BUY* and *SUM_SELL* signals and *NET_BUY* and *NET_SELL* signals. SD denotes the standard deviations. Various percentiles are reported. Using the buy and sell lists, *SUM_BUY* (*SUM_SELL*) is the sum of *BUY* (*SELL*) signals across the lists. *NET_BUY* and *NET_SELL* signals are obtained by netting the *SUM_BUY* and *SUM_SELL* signals. *NET_BUY* has a value when *SUM_BUY* is greater than *SUM_SELL*, and *NET_SELL* has a value when *SUM_SELL* is greater than *SUM_BUY*. We treat zero *NET_BUY* or *NET_SELL* as missing observations. Panel B also reports a detailed distribution of *N_TWTR*, which is the number of RIC information that appeared in the previous 24 hours. A total of 735 unique stocks appeared in some tweets having RIC information.

Panel A. Full Sample

Variable	N	Mean	SD	Min	p1	p5	p10	p25	p50	p75	p90	p95	p99	Max
BUY	1059956	0.18	0.38	0	0	0	0	0	0	0	1	1	1	1
SELL	1059956	0.24	0.43	0	0	0	0	0	0	0	1	1	1	1
D_TWTR	1059956	0.01	0.09	0	0	0	0	0	0	0	0	0	0	1

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel B. Detailed Distribution

Variable	N	Mean	SD	Min	p1	p5	p10	p25	p50	p75	p90	p95	p99	Max
SUM_BUY	190130	6.02	11.67	1	1	1	1	1	2	5	17	29	56	143
SUM_SELL	251410	3.15	4.43	1	1	1	1	1	1	3	8	12	22	68
NET_BUY	119880	5.54	10.13	1	1	1	1	1	1	5	16	26	49	117
NET_SELL	163508	1.89	2.21	1	1	1	1	1	1	2	3	5	12	45
N_TWTR	8229	2.72	3.39	1	1	1	1	1	2	3	6	8	16	74

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 3. Stock-Level Characteristics and ChatGPT Day Trading Signals

This table reports panel regression results of regressing ChatGPT day trading signals on various stock-level characteristics. The main dependent variables are *BUY* and *SELL* signals. *BUY* (*SELL*) signal is a dummy variable having 1 when a stock is selected at least once as a stock buy (sell). Panel A uses stock-level characteristics as the main independent variables. *SIZE* is the log of market capitalization. *MOM1M* is the previous 20 days of returns. *MOM12M* is the previous 12 to 1-month returns. $RET\ (t-1)$ is the previous day's returns. Panel B uses an indicator for RIC information (*D_TWTR*) and various market information as the main independent variables. *D_TWTR* for a stock is a dummy variable having 1 one when the RIC information of the stock is tagged in a tweet, which is provided by Refinitiv Eikon News Monitor. *N_TWTR* for a stock is the total number of tweets in which the stock appears. *D_SP500* is a dummy variable having 1 one a stock is a constituent of the S&P 500 index. *D_NASDAQ* is a dummy variable having 1 one a stock is traded in NASDAQ. Panel C uses trading frictions as the main independent variables. *TURN* is the previous trading day's share turnover. *ILLIQ* is the Amihud (2002) illiquidity measure multiplied by 10,000. *BASPREAD* is the bid-ask spread, and *RETVOL* is the previous 20 days of return volatility. Panel D uses earnings announcement day as the main independent variable. Earnings announcement days are collected from COMPUSTAT. *D_RDQ* is a dummy variable having 1 when a firm announced earnings in the previous 7 days. All independent variables are lagged by one day. All continuous independent variables are winsorized at 1% and 99% level. All estimations include date-fixed effects. The standard errors are two-way clustered by stock- and day-level, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel A. Stock-Level Characteristics

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	BUY					*SELL*				
SIZE	0.072***				0.085***	0.066***				0.079***
	(41.38)				(45.40)	(39.56)				(44.49)
MOM1M		0.063***			-0.112***		0.046***			-0.116***
		(6.81)			(-15.02)		(5.09)			(-13.58)
MOM12M			0.053***		-0.138***			0.041***		-0.136***
			(8.04)		(-21.39)			(5.90)		(-19.78)
$RET\ (t-1)$				0.107***	-0.017				0.048*	-0.055**
				(3.70)	(-0.71)				(1.68)	(-2.21)
Observations	1059956	1059956	1034107	1059956	1034107	1059956	1059956	1034107	1059956	1034107
R-sq	0.223	0.007	0.011	0.007	0.249	0.159	0.009	0.011	0.009	0.177

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel B. Tweets and Market Information

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	BUY					*SELL*				
D_TWTR	0.666***				0.231***	0.623***				0.224***
	(35.61)				(15.50)	(36.23)				(15.71)
N_TWTR		0.103***			-0.002		0.095***			-0.001
		(8.81)			(-0.95)		(8.80)			(-0.42)
D_SP500			0.621***		0.438***			0.571***		0.399***
			(47.83)		(29.29)			(48.62)		(27.80)
D_NASDAQ				-0.127***	0.018***				-0.152***	-0.024***
				(-13.04)	(2.62)				(-15.56)	(-3.11)
SIZE					0.039***					0.033***
					(22.37)					(18.34)
Observations	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956
R-sq	0.030	0.017	0.271	0.032	0.316	0.025	0.016	0.191	0.038	0.222

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel C. Trading Frictions

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	BUY					*SELL*				
TURN	1.179***				1.815***	1.661***				2.353***
	(6.15)				(12.33)	(8.22)				(14.35)
ILLIQ		-6.426***			5.670***		-6.638***			4.568***
		(-22.07)			(20.30)		(-21.18)			(15.63)
BASPREAD			-1.436***		0.242***			-1.306***		0.116***
			(-22.79)		(6.87)			(-20.04)		(2.95)
RETVOL				-2.165***	0.853***				-1.938***	0.724***
				(-20.12)	(13.19)				(-17.37)	(9.96)
SIZE					0.089***					0.080***
					(42.58)					(40.18)
Observations	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956
R-sq	0.010	0.023	0.032	0.030	0.248	0.014	0.023	0.026	0.024	0.179

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel D. Earnings Announcement

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	BUY					*SELL*				
D_RDQ	0.009	0.005	0.003	0.003	-0.010***	0.004	0.001	-0.001	-0.005	-0.017***
	(1.07)	(1.40)	(0.39)	(0.29)	(-2.62)	(0.56)	(0.20)	(-0.19)	(-0.66)	(-4.21)
SIZE		0.072***			0.072***		0.066***			0.067***
		(40.87)			(41.70)		(39.08)			(40.41)
D_TWTR			0.666***		0.344***			0.623***		0.319***
			(35.57)		(23.82)			(36.21)		(22.53)
TURN				1.177***	2.162***				1.665***	2.581***
				(6.12)	(15.53)				(8.20)	(16.93)
Observations	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956
R-sq	0.006	0.221	0.030	0.010	0.240	0.009	0.158	0.025	0.014	0.176

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 4. Intraday Portfolio Performance of ChatGPT Day Trading Signals

This table reports the intraday excess returns and CAPM alpha of the day trading strategy based on ChatGPT day trading signals. The sample period is from December 2022 to December 2023, which is 271 trading days. Panel A reports portfolio returns by the *BUY* and *SELL* signals and their difference, denoted by *BMS* (*BUY* minus *SELL*). The *BUY* (*SELL*) portfolio is constructed by equally weighting stocks with *BUY* (*SELL*) signal across the lists. We hold stocks from the market opening to closing. Accordingly, our portfolio return is open-to-close price returns. We proxy risk-free rates using open-to-close price returns of BIL (SPDR Bloomberg 1-3 Month T-Bill ETF). The excess return is the difference between portfolio returns and risk-free rates. CAPM alpha is estimated using market factor proxied by the open-to-close price return of SPY (SPDR S&P 500 ETF Trust) return minus BIL return. The excess return and CAPM alpha are reported in monthly percentage terms. Panel B reports portfolio returns constructed using *SUM_BUY* and *SUM_SELL* signals, and their difference is denoted as *SUM_BMS*. *SUM_BUY* (*SUM_SELL*) portfolio is constructed by weighting stocks using *SUM_BUY* (*SUM_SELL*) signal. Panel C reports portfolio returns constructed using the *NET_BUY* and *NET_SELL* signals, and their difference is denoted by *NET_BMS*. *NET_BUY* (*NET_SELL*) portfolio is constructed by weighting stocks using *NET_BUY* (*NET_SELL*) signal. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. *BUY* and *SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
BUY	*SELL*	*BMS*	*BUY*	*SELL*	*BMS*
-0.098	-0.644	0.546**	-1.448**	-1.973**	0.525**
(-0.07)	(-0.44)	(2.27)	(-2.02)	(-2.35)	(2.19)

Panel B. *SUM_BUY* and *SUM_SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
SUM_BUY	*SUM_SELL*	*SUM_BMS*	*SUM_BUY*	*SUM_SELL*	*SUM_BMS*
1.585	0.073	1.512***	0.178	-1.450*	1.629***
(1.17)	(0.04)	(2.84)	(0.34)	(-1.65)	(3.17)

Panel C. *NET_BUY* and *NET_SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
1.850	-1.663	3.513***	0.524	-3.193**	3.717***
(1.48)	(-0.90)	(3.16)	(1.15)	(-2.58)	(3.42)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 5. Excluding Twitter-Mentioned Firms on Twitter

This table reports the intraday performance of the day trading strategy using subsamples of stocks separated by whether or not they are mentioned in Twitter news articles. A stock is classified as Twitter-mentioned when the stock's Refinitiv Instrument Code (RIC) is tagged in some tweets. RIC is provided by Refinitv News Monitor. All portfolios are constructed by the *NET_BUY* and *NET_SELL* signals. *NET_BMS* is the difference between portfolio returns constructed by *NET_BUY* and *NET_SELL* signals. Panel A uses a subsample of stocks by excluding the Twitter-mentioned firms. Panel B uses a subsample of stocks that are Twitter-mentioned. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. Excluding Twitter-Mentioned Firms

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
1.946	-1.724	3.670***	0.626	-3.241***	3.868***
(1.54)	(-0.95)	(3.38)	(1.24)	(-2.64)	(3.63)

Panel B. Using Only Twitter-Mentioned Firms

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
1.425	0.115	1.333	0.078	-2.060	2.132
(1.04)	(0.02)	(0.20)	(0.12)	(-0.33)	(0.34)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 6. Time-Series Portfolio Performance by the Intensity of Firm-Specific Information

This table reports the time-series regression results of examining the relationship between intraday portfolio performance and the intensity of firm-specific information. The main dependent variable is excess portfolio returns. Column (1) uses portfolio returns constructed by the *NET_BUY* signal, and Column (2) uses the *NET_SELL* signal. Column (3) uses NET_BMS, which is the difference between *NET_BUY* and *NET_SELL*. The main independent variable is *D_TWTR_HIGH*, which is a dummy variable having 1 when a day has above the median percentage of firm-specific Twitter news articles. A Twitter news article is categorized as having firm-specific information when it is tagged with Refinitiv Instrument Code (RIC), which is provided by Refinitiv News Monitor. The percentage of firm-specific Twitter news articles is the percentage of news articles tagged with the RIC code. *MKTRF* is the open-to-close market factor return. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

	(1) *NET_BUY*	(2) *NET_SELL*	(3) *NET_BMS*
D_TWTR_HIGH	-1.537*	1.062	-2.599
	(-1.74)	(0.41)	(-1.14)
MKTRF	1.243***	1.443***	-0.201***
	(42.87)	(24.74)	(-3.47)
Constant	1.301*	-3.730**	5.031***
	(1.92)	(-2.06)	(3.50)
Observations	271	271	271
R-sq	0.864	0.558	0.029

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 7. The Effects of Firm-Specific News in Input Prompts

This table reports the portfolio performance conditional on using firm-specific news as input prompts. If a tweet contains some firm-specific information, then Eikon provides relevant best-matching Refinitiv Instrument Code (RIC). *Firm-Specific News* indicates tweets having RIC information. All portfolio returns are open-to-close price returns. All portfolio performance is CAPM alpha, which is reported in monthly percentage terms. Panel A shows the benchmark portfolio performance using all news to generate ChatGPT signals. Panel B shows the performance using *NET_BUY* and *NET_SELL* signals generated by using only *Firm-Specific News* as input prompts to generate ChatGPT signals. Panel C shows the performance using *NET_BUY* and *NET_SELL* signals generated by excluding *Firm-Specific News* as input prompts to generate ChatGPT signals. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. All News

NET_BUY	*NET_SELL*	*NET_BMS*
0.524	-3.193**	3.717***
(1.15)	(-2.58)	(3.42)

Panel B. Only *Firm-Specific News*

NET_BUY	*NET_SELL*	*NET_BMS*
0.090	-2.452**	2.542***
(0.18)	(-2.41)	(2.88)

Panel C. Excluding *Firm-Specific News*

NET_BUY	*NET_SELL*	*NET_BMS*
0.434	-3.486***	3.920***
(1.00)	(-2.85)	(3.71)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 8. Topics for the Reason for Selecting the Buy and Sell Lists

This table reports topics for the reason for selecting the buy and sell lists that ChatGPT provides. The reasons are directly obtained from ChatGPT. The procedure is as follows. For each batch and random seed for 30 iterations, after obtaining the lists of stock tickers to buy and sell from ChatGPT, I ask ChatGPT again for the reason for selecting those stocks for each buy and sell list. For buy and sell reasons, each topic is modeled using the Latent Dirichlet Allocation (LDA) model. The number of topics for each buy and sell list is 1. The column *Topic for BUY Reason* reports the top 20 LDA-based keywords selected by their probability in the topic of the buy reasons. Similarly, *Topic for SELL Reason* reports the top 20 LDA-based keywords for the topic of the sell reasons. All keywords are lemmatized and lowercase by the Python Natural Language Toolkit (NLTK).

Order	*Topic for BUY Reason*		*Topic for SELL Reason*	
	Keyword	Probability	Keyword	Probability
1	include	0.0186	include	0.0203
2	tech	0.0180	due	0.0176
3	strong	0.0159	such	0.0174
4	development	0.0156	impact	0.0169
5	such	0.0144	sector	0.0164
6	apple	0.0143	industry	0.0139
7	aapl	0.0139	regulatory	0.0134
8	amazon	0.0128	decline	0.0117
9	additionally	0.0128	issue	0.0116
10	likely	0.0120	uncertainty	0.0114
11	amzn	0.0119	performance	0.0111
12	microsoft	0.0116	base	0.0106
13	trend	0.0106	concern	0.0101
14	advancement	0.0101	risk	0.0096
15	msft	0.0100	financial	0.0094
16	healthcare	0.0100	lead	0.0093
17	tsla	0.0099	economic	0.0087
18	base	0.0098	energy	0.0086
19	tesla	0.0096	consumer	0.0081
20	energy	0.0092	change	0.0081

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 9. Economic Link Between the Generated Stock Tickers

This table reports the analysis results of examining the economic link between the generated stock tickers. Firms with *BUY* and *SELL* signals are matched with stocks that are not selected by ChatGPT neither as buy nor sell list. Matching variables include size, share turnover, a dummy for the Twitter-mentioned, past 12-month momentum, industry momentum, a dummy for earnings announcement within 7 days, and indicators of SIC 2-digit codes. I start matching each buy and sell list from the largest market value firm in the list without replacement. Panel A compares the industry-level concentration between the stocks in the ChatGPT-generated lists and their matched lists of stocks. Columns (1) to (4) report the results for the buy list consisting of stocks having the *NET_BUY* signal. Industry-level concentration is measured by the Herfindahl-Hirschman Index (HHI). HHI is calculated for each day. Each day, I calculate HHI using the SIC code. Column (1) uses SIC1, which is the 1-digit SIC code, to calculate HHI for the buy list and its matching group. The main dependent variable is the difference between the HHI of the buy list and the HHI of the matching group. The time-series difference of HHI is regressed on the constant term to estimate the time-series average of the difference. Columns (2), (3), and (4) use SIC2, SIC3, and SIC4, which are 2-, 3-, and 4-digit SIC codes to calculate HHI, respectively. Columns (5) to (8) report the results for the sell lists consisting of stocks having the *NET_SELL* signal. Panel B compares the proportion of stocks that are linked as supply chains for the ChatGPT-generated lists and their matched lists of stocks. Columns (1) to (4) report the results for the buy list consisting of stocks having the *NET_BUY* signal. The supply chain data is obtained from the Compustat Segment. I calculated the proportion of stocks that are linked through the supply chain in each of the buy list and its matching group. Column (1) uses supply chain data ranging from 2010 to 2022 (2010s). The main dependent variable is the difference between the proportion of firms linked through the supply chain in the buy list and their matching group. The time-series difference of the proportion of supply chain linkage is regressed on the constant term to estimate the time-series average of the difference. Columns (2), (3), and (4) uses supply chain data covering the 2000s, 1990s, and 1980s, respectively. Columns (5) to (8) report the results for the sell lists consisting of stocks having the *NET_SELL* signal. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel A. Industry-Level Concentration

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	NET_BUY				*NET_SELL*			
	SIC1	SIC2	SIC3	SIC4	SIC1	SIC2	SIC3	SIC4
	Difference of HHI				Difference of HHI			
Constant	-0.016***	-0.025***	-0.028***	-0.036***	-0.034***	-0.046***	-0.046***	-0.046***
	(-31.14)	(-37.04)	(-38.93)	(-50.66)	(-43.32)	(-55.03)	(-52.66)	(-51.84)
Observations	271	271	271	271	271	271	271	271

Panel B. Supply Chain

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	NET_BUY				*NET_SELL*			
	2010s	2000s	1990s	1980s	2010s	2000s	1990s	1980s
	Differences in Supply Chain Linkage				Differences in Supply Chain Linkage			
Constant	0.128***	0.083***	0.044***	0.014***	0.052***	0.023***	-0.000	-0.009***
	(107.88)	(74.11)	(58.07)	(23.80)	(57.38)	(37.01)	(-0.50)	(-23.68)
Observations	271	271	271	271	271	271	271	271

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 10. Economic Link and Portfolio Performance

This table reports various portfolio performances to examine the economic mechanism of ChatGPT signal generation. All portfolio returns are reported in CAPM alpha, which is converted into monthly percentage terms. *NET_BUY* (*NET_SELL*) is the CAPM alpha of the portfolio constructed by the *NET_BUY* (*NET_SELL*) signal. *NET_BMS* is the CAPM alpha of the difference in portfolio returns between *NET_BUY* and *NET_SELL*. Panel A reports the results for portfolios constructed by subsamples of stocks classified by within-industry signal. For each day and each SIC 2-digit code, a stock is classified as the *High* group if its *NET_BUY* or *NET_SELL* signal is above the median of the group and as the *Low* group when the signals are below the median. Panel B reports the results for portfolios constructed by subsamples of stocks classified by the across-industry signal. For each day, the across-industry signal is obtained by constructing SIC 2-digit-level portfolios and calculating the median of the *NET_BUY* and *NET_SELL* signals of the industry portfolios. Then, a stock is classified as the *High* group when the industry that includes the stock has the above the median *NET_BUY* or *NET_SELL* signal and as the *Low* group when below the median. Panel C reports the results of portfolios constructed by subsamples of stocks linked through the supply chain links or not. Supply chain data is obtained from the Compustat Segment. For each day and each of the stocks with the *NET_BUY* signal, stocks are classified as the *Yes* group when they have supply chain customers among other stocks with the *NET_BUY* signal and as the *No* group otherwise. Stocks with the *NET_SELL* signal are similarly categorized as the *Yes* or *No* group using supply chain data. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. Within-Industry Signal

High			*Low*		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
0.823*	-3.922***	4.745***	-0.829	-2.365**	1.536**
(1.73)	(-2.60)	(3.37)	(-1.24)	(-2.34)	(2.46)

Panel B. Across-Industry Signal

High			*Low*		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
0.599	-3.463**	4.062***	-1.070	-2.016**	0.946
(1.27)	(-2.51)	(3.24)	(-1.41)	(-2.10)	(1.46)

Panel C. Supply Chain

Yes			*No*		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
-0.048	-2.336**	2.288*	0.585	-3.236**	3.821***
(-0.08)	(-2.17)	(1.92)	(1.21)	(-2.56)	(3.50)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 11. High-Frequency Portfolio Performance

This table reports the 30-minute interval risk-adjusted returns of strategies using ChatGPT day trading signals. Portfolios on day t are constructed by *NET_BUY* and *NET_SELL* signals, which are produced at 9:30 AM of day t using Twitter news articles of the previous 24 hours. *NET_BMS* is the difference between *NET_BUY* and *NET_SELL* portfolio returns. The 30-minute interval of returns is calculated using the quote midpoint returns calculated from NYSE Trade and Quote (TAQ). The column labeled *From* indicates the start of the 30-minute interval, and *To* indicates the end of the interval. Overnight is from the close price of day t to the opening price of day $t + 1$. CAPM Alpha is measured using a 30-minute market factor proxied by SPY (SPDR S&P 500 ETF Trust) and BIL (SPDR Bloomberg 1-3 Month T-Bill ETF). The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

		CAPM Alpha			t-stat		
From	*To*	*NET_BUY*	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
9:30	10:00	0.848	-5.678**	6.527***	0.599	-2.051	2.752
10:00	10:30	-0.553	-1.582	1.041	-0.677	-0.870	0.660
10:30	11:00	-0.187	-1.658	1.467	-0.264	-1.434	1.382
11:00	11:30	1.395***	0.212	1.185	2.953	0.180	1.239
11:30	12:00	0.404	-1.871**	2.291***	1.017	-2.021	2.691
12:00	12:30	-0.172	-0.247	0.080	-0.418	-0.269	0.105
12:30	13:00	0.152	-0.922	1.088*	0.462	-1.301	1.740
13:00	13:30	0.575	0.455	0.104	1.548	0.534	0.136
13:30	14:00	0.006	-0.594	0.606	0.018	-0.876	0.981
14:00	14:30	0.961**	0.925	0.028	2.444	1.104	0.038
14:30	15:00	-0.610**	-0.734	0.106	-1.985	-0.966	0.168
15:00	15:30	0.434	-0.503	0.949	1.132	-0.641	1.449
15:30	16:00	-1.384***	-3.172***	1.788**	-3.875	-3.179	1.990
Overnight		0.486	7.954***	-7.429***	0.433	3.114	-3.062

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 12. Overnight and One-day Portfolio Performance of ChatGPT Day Trading Signals

This table reports the overnight CAPM alpha of portfolios constructed by the ChatGPT day trading signal. To examine the performance of signals generated before the market opening at day t, I report two types of portfolio returns: holding stocks from day t's closing to the next trading day's opening (*Close-to-Open*) and holding stocks from day t's opening to the next trading day's opening (*Open-to-Open*). We proxy the overnight risk-free rates using close-to-open price returns of BIL (SPDR Bloomberg 1-3 Month T-Bill ETF) and market returns using close-to-open price returns of SPY (SPDR S&P 500 ETF Trust). Panel A reports overnight portfolio performance of *BUY* and *SELL* signals. Panel B reports for *SUM_BUY* and *SUM_SELL* signals. Panel C reports for *NET_BUY* and *NET_SELL* signals. All CAPM alphas are reported in monthly percentage terms. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. *BUY* and *SELL* Signal

Close-to-Open			*Open-to-Open*		
BUY	*SELL*	*BMS*	*BUY*	*SELL*	*BMS*
0.659**	1.109***	-0.450***	-0.779	-0.938	0.159
(2.27)	(3.44)	(-3.00)	(-0.93)	(-0.96)	(0.56)

Panel B. *SUM_BUY* and *SUM_SELL* Signal

Close-to-Open			*Open-to-Open*		
SUM_BUY	*SUM_SELL*	*SUM_BMS*	*SUM_BUY*	*SUM_SELL*	*SUM_BMS*
0.038	0.717*	-0.679***	0.247	-0.729	0.976
(0.14)	(1.91)	(-2.86)	(0.41)	(-0.69)	(1.52)

Panel C. *NET_BUY* and *NET_SELL* Signal

Close-to-Open			*Open-to-Open*		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
0.024	1.698***	-1.674***	0.569	-1.549	2.118
(0.09)	(3.36)	(-3.34)	(1.11)	(-1.03)	(1.59)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 13. Overnight Retail Order Imbalance and ChatGPT Day Trading Signals

This table reports the panel regression results of analyzing the relationship between the overnight order imbalance by retail investors and ChatGPT day trading signals. The main dependent variable is the *Overnight Order Imbalance*. Using TAQ data, order direction is measured by following Lee and Ready (1991). *Overnight Order Imbalance* on day t is the difference between buying and selling orders divided by the sum of buying and selling orders during the market closing at day t to the market opening of day $t + 1$. Retail investors' trading is measured by following Boehmer et al. (2021). Column (1) uses all trades to calculate *Overnight Order Imbalance*. Column (2) uses retail trades to calculate *Overnight Order Imbalance*, which is the difference between retail buying and selling orders divided by the sum of retail buying and selling orders. Column (3) uses non-retail trades to calculate *Overnight Order Imbalance*. Columns (1), (2) and (3) use the full sample to run the panel regressions. Columns (4), (5), and (6) repeat Columns (1), (2), and (3) using stocks having non-zero *NET_BUY* or *NET_SELL* signals. The main independent variables are *NET_BUY* and *NET_SELL* signals, which are constructed at the market opening on day t. *D_NET_SELL* is a dummy variable having 1 when the *NET_SELL* signal is non-zero. The row labeled as *F* indicates the F-statistics is the test for coefficient estimations for *NET_BUY* and *NET_SELL* are equal, and the *p-value* is the p-value for the F statistics. Control variables include the log of market capitalization, share turnover, an indicator for the tweet mentioned, the absolute value of the previous day's returns, retail net buying (Berkman et al. (2012)), illiquidity, bid-ask spread, return volatility, stock return momentum of previous 2 to 12 months, and previous month's returns, and previous day's returns, and SIC 2-digit industry returns. All continuous variables are winsorized at 1% and 99% each day. All specifications include date-fixed effects. The standard errors are two-way clustered by stock- and day-level, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

	(1)	(2)	(3)	(4)	(5)	(6)
Sample	*Full Sample*			*NET_BUY*>0 or *NET_SELL*>0		
	All	*Retail*	*Non-Retail*	*All*	*Retail*	*Non-Retail*
	Overnight Order Imbalance					
NET_BUY	-0.003***	0.013***	-0.003***	-0.002**	0.010***	-0.002**
	(-3.13)	(12.72)	(-3.10)	(-2.53)	(10.17)	(-2.47)
NET_SELL	-0.000	0.029***	-0.000	-0.000	0.023***	-0.000
	(-0.08)	(8.52)	(-0.06)	(-0.15)	(7.76)	(-0.10)
D_NET_SELL	-0.004	-0.035***	-0.004	-0.007	-0.042***	-0.007
	(-0.88)	(-5.05)	(-0.89)	(-1.29)	(-4.81)	(-1.26)
Observations	981297	371325	981266	273745	150209	273740
R-sq	0.030	0.027	0.030	0.033	0.039	0.033
F	1.213	26.962	1.235	0.628	23.084	0.670
p-value	0.272	0.000	0.267	0.429	0.000	0.414
Controls	Y	Y	Y	Y	Y	Y

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 14. Intraday and Overnight Portfolio Performance Conditional on ChatGPT Signals Using Each of *Intraday News* and *Overnight News*

This table reports the intraday and overnight CAPM alpha by using several ChatGPT signals generated by using different sets of tweets. *Intraday Return* on day t indicates portfolio returns by holding stocks from 9:30 AM to 4:00 PM of day t. *Overnight Return* at day t indicates portfolio returns by holding stocks from 4:00 PM of day t to the next trading day's market opening at 9:30 AM. *Intraday News* at day t indicates tweets posted from 9:30 AM to 4:00 PM. *Overnight News* on day $t-1$ indicates tweets posted from 4:00 PM of day $t-1$ to the next trading day t's market opening at 9:30 AM. All portfolios are constructed by *NET_BUY* and *NET_SELL* signals. All CAPM alphas are reported in monthly percentage terms. Panel A reports for *Overnight Return* at day t using ChatGPT signals by *Intraday News* at day t. Panel B reports for *Overnight Return* at day t using ChatGPT signals by *Overnight News* at day $t-1$. Panel C reports for *Intraday Return* at day t using ChatGPT signals by *Overnight News* at day $t-1$. Panel D reports the CAPM alpha for *Intraday Return* at day t using ChatGPT signals based only on *Intraday News* at day $t-1$. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. *Overnight Return* by *Intraday News*

NET_BUY	*NET_SELL*	*NET_BMS*
-0.038	1.488***	-1.526***
(-0.14)	(2.80)	(-3.07)

Panel B. *Overnight Return* by *Overnight News*

NET_BUY	*NET_SELL*	*NET_BMS*
-0.021	1.444***	-1.465***
(-0.08)	(3.00)	(-2.88)

Panel C. *Intraday Return* by *Overnight News*

NET_BUY	*NET_SELL*	*NET_BMS*
0.202	-2.920**	3.122***
(0.42)	(-2.45)	(2.98)

Panel D. *Intraday Return* by *Intraday News*

NET_BUY	*NET_SELL*	*NET_BMS*
0.551	-3.214***	3.765***
(1.16)	(-2.64)	(3.70)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 15. Overnight Portfolio Performance Using Contemporaneous News

This table reports the overnight portfolio performance of ChatGPT signals generated by using contemporaneous overnight news. Overnight returns in this analysis are measured by holdings stocks from market close (4:00 PM) at day t to market open (9:30 AM) at the next trading day (*Close-to-Open*). Overnight news is tweets posted from market close (4:00 PM) at day t to market open (9:30 AM) at the next trading day. Panel A uses *BUY* and *SELL* signals to construct portfolios. Panel B uses *SUM_BUY* and *SUM_SELL* signals. Panel C uses *NET_BUY* and *NET_SELL* signals. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. *BUY* and *SELL* Signal

Overnight Excess Return (monthly, %)			Overnight CAMP Alpha (monthly, %)		
BUY	*SELL*	*BMS*	*BUY*	*SELL*	*BMS*
0.811	0.946	-0.135	0.539*	0.674**	-0.135
(1.01)	(1.16)	(-0.91)	(1.84)	(2.05)	(-0.91)

Panel B. *SUM_BUY* and *SUM_SELL* Signal

Overnight Excess Return (monthly, %)			Overnight CAMP Alpha (monthly, %)		
SUM_BUY	*SUM_SELL*	*SUM_BMS*	*SUM_BUY*	*SUM_SELL*	*SUM_BMS*
0.308	0.453	-0.145	0.019	0.152	-0.133
(0.37)	(0.50)	(-0.54)	(0.07)	(0.38)	(-0.50)

Panel C. *NET_BUY* and *NET_SELL* Signal

Overnight Excess Return (monthly, %)			Overnight CAMP Alpha (monthly, %)		
NET_BUY	*NET_SELL*	*NET _BMS*	*NET_BUY*	*NET_SELL*	*NET _BMS*
0.427	0.924	-0.498	0.147	0.630	-0.483
(0.53)	(0.96)	(-0.97)	(0.54)	(1.23)	(-0.93)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 16. Day Trader, Swing Trader, and Trader

This table reports the portfolio performance of ChatGPT day trading signals generated using different prompts. Panel A shows the benchmark performance of asking ChatGPT to pretend to be a *Day Trader*. Panel B shows the performance of asking to be a *Swing Trader*. Panel C shows the performance of asking to be a *Trader*, which is asking without any specific trading style. All portfolios are constructed by *NET_BUY* and *NET_SELL* signals. All portfolio returns are CAPM alpha reported in monthly percentage terms. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. Day Trader

Intraday Return (*Open-to-Close*)			*Overnight Return* (*Close-to-Open*)			One-day Return (*Open-to-Open*)		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
0.524	-3.193**	3.717***	0.024	1.698***	-1.674***	0.569	-1.549	2.118
(1.15)	(-2.58)	(3.42)	(0.09)	(3.36)	(-3.34)	(1.11)	(-1.03)	(1.59)

Panel B. Swing Trader

Intraday Return (*Open-to-Close*)			*Overnight Return* (*Close-to-Open*)			One-day Return (*Open-to-Open*)		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
0.470	-3.624***	4.093***	-0.063	1.575***	-1.637***	0.440	-2.066	2.506**
(1.16)	(-3.13)	(4.14)	(-0.25)	(3.25)	(-3.36)	(0.98)	(-1.46)	(2.03)

Panel C. Trader

Intraday Return (*Open-to-Close*)			*Overnight Return* (*Close-to-Open*)			One-day Return (*Open-to-Open*)		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
0.305	-3.171***	3.477***	-0.126	1.606***	-1.732***	0.206	-1.596	1.801
(0.74)	(-2.70)	(3.48)	(-0.49)	(3.24)	(-3.34)	(0.42)	(-1.09)	(1.39)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 17. Decay of ChatGPT Day Trading Signals

This table reports the portfolio performances constructed by lagged ChatGPT day trading signals. We use *NET_BUY* and *NET_SELL* signals to construct portfolios. Each row indicates the lag between the portfolio construction and *NET_BUY* and *NET_SELL* signals. $t - i$ indicates i days of lags. Intraday excess return is open-to-close price returns minus corresponding risk-free rates. Overnight excess return is close-to-open price returns minus corresponding risk-free rates. All portfolio returns are CAPM alpha reported in monthly percentage terms. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

	Intraday CAPM Alpha (monthly, %)			Overnight CAPM Alpha (monthly, %)		
	NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
$t - 1$	0.457	-2.944**	3.402***	0.024	1.698***	-1.674***
	(1.00)	(-2.28)	(2.99)	(0.09)	(3.36)	(-3.34)
$t - 2$	0.223	-3.197***	3.420***	-0.068	1.609***	-1.677***
	(0.49)	(-2.73)	(3.52)	(-0.25)	(3.24)	(-3.37)
$t - 3$	0.717	-3.512***	4.229***	-0.028	1.583***	-1.612***
	(1.54)	(-2.93)	(4.00)	(-0.11)	(3.21)	(-3.10)
$t - 4$	0.255	-2.989**	3.244***	0.041	1.399***	-1.358***
	(0.56)	(-2.53)	(3.30)	(0.15)	(2.82)	(-2.74)
$t - 5$	0.465	-3.406***	3.871***	-0.201	1.694***	-1.895***
	(1.00)	(-2.86)	(3.71)	(-0.68)	(3.53)	(-3.88)

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 18. Important Keywords, Their Persistentcy, and Impacts on the Day Trading Strategy by ChatGPT Signals

This table reports the time-series regression results of regressing ChatGPT-based intraday portfolio returns on the scores of importance for various keywords. Portfolio returns are open-to-close intraday returns of *NET_BUY*, *NET_SELL*, and *NET_BMS* portfolios. Keywords are selected from their Term Frequency-Inverse Document Frequency (TF-IDF) score. There are several steps to obtain TF-IDF. I start by processing raw text data using the Python Natural Language Toolkit (NLTK). I classify words using Part-of-Speech (POS) tagging. I use only nouns, verbs, adjectives, and adverbs. I also lemmatize all words. I aggregate tweets at the daily level, which results in 271 documents. Using 271 documents, I calculate TF-IDF using N-grams ranging from 2 to 10 by imposing the minimum document frequency to 20 times. The final selected number of keywords is 11,555. I calculate the time-series average TF-IDF score for each keyword and select the first 1,000 keywords from the highest time-series average TF-IDF score. The column labeled as *Rank* is the rank of the TF-IDF score among 1,000 keywords. Using the TF-IDF values of these keywords, I report the auto-regressive coefficient of each keyword. I run the following regression:

$$Keyword_t = Constant + AR \cdot Keyword_{t-1} + \epsilon_t.$$

$Keyword_t$ is the TF-IDF value of each keyword at day t. I report the estimation results for AR in the last column labeled by *AR Coef*. To examine the explanatory power of each keyword for ChatGPT-based portfolio performance, I run the following regressions.

$$Intraday\ Excess\ Returns_t = Constant + \beta_{MKTRF} \cdot MKTRF_t + \beta_{Keyword} \cdot H\{Keyword_{t-1}\} + \epsilon_t.$$

$H\{Keyword_{t-1}\}$ is a dummy variable having 1 when the keyword's TF-IDF value is greater than its median. $Intraday\ Excess\ Returns_t$ can be one of *NET_BUY*, *NET_SELL*, and *NET_BMS*. Portfolio returns are included in the regression after being converted into monthly percentage terms. The column labeled *Keyword* reports the selected keywords, which are all lowercase by NLTK. The column labeled *N* reports the number of observations having a TF-IDF value greater than 0 among 271 days. Panel A reports the results for the top 30 TF-IDF score keywords, and Panel B reports the results for the top 30 significant $\beta_{Keyword}$ p-value. For brevity, I only report $\beta_{Keyword}$ for time-series regressions. The standard errors are estimated using the Newey-West adjustment. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel A. Top 30 TF-IDF				$\beta_{Keyword}$		
Keyword	*Rank*	*N*	*AR Coef*	*NET_BUY*	*NET_SELL*	*NET_BMS*
interest rate	1	271	0.331***	-0.312	0.255	-0.566
central bank	2	271	0.207***	-1.868**	-2.977	1.109
new york	3	271	0.326***	0.776	0.471	0.305
wall street	4	271	0.277***	-0.244	0.553	-0.797
prime minister	5	270	0.336***	1.116	1.304	-0.188
big take	6	271	0.365***	-0.509	-0.145	-0.364
last year	7	262	0.611***	-0.521	-2.480	1.958
elon musk	8	265	0.515***	1.037	0.609	0.429
donald trump	9	265	0.448***	-0.868	-3.474	2.606
first time	10	269	0.077	-0.457	-0.175	-0.282
next year	11	258	0.596***	1.535*	4.399*	-2.863
latest update	12	182	0.666***	0.068	-0.732	0.800
hong kong	13	266	0.254***	0.934	6.903***	-5.969***
credit suisse	14	222	0.664***	-0.741	-3.073	2.332
bankman fried	15	175	0.713***	0.776	2.380	-1.604
source say	16	266	0.065	0.630	2.926	-2.296
rishi sunak	17	245	0.191***	1.119	-0.914	2.033
sam bankman	18	173	0.712***	0.574	2.935	-2.361
sam bankman fried	19	173	0.704***	0.574	2.935	-2.361
debt ceiling	20	93	0.868***	0.541	-3.123	3.664
read big	21	270	0.259***	0.042	-2.321	2.363
read big take	22	270	0.256***	0.045	-2.360	2.405
federal reserve	23	255	0.264***	0.218	-0.270	0.488
real estate	24	262	0.207***	0.258	2.810	-2.552
rate hike	25	242	0.428***	-0.308	1.563	-1.872
israel hamas	26	57	0.872***	0.661	2.703	-2.042
need know	27	267	0.251***	-1.365*	1.372	-2.737
joe biden	28	250	0.307***	1.235	-0.897	2.132
south africa	29	255	0.251***	-0.126	-1.200	1.074
bank england	30	230	0.519***	0.430	3.190	-2.760

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel B. Top 30 Significance

				$\beta^{D}_{Keyword}$		
Keyword	*Rank*	*N*	*AR Coef*	*NET_BUY*	*NET_SELL*	*NET_BMS*
activision blizzard	495	71	0.373***	-1.051	-8.786***	7.736***
energy price	733	99	0.076	0.306	-6.521***	6.828***
year prison	528	105	0.227**	0.159	-6.017**	6.177***
treasury secretary janet yellen	189	122	0.386***	0.785	-5.501**	6.286***
treasury secretary janet	190	122	0.386***	0.785	-5.501**	6.286***
bank canada	398	113	0.278***	-0.812	-6.228***	5.416***
chinese president xi jinping	625	92	0.305***	1.069	-5.016**	6.085***
10 year	690	89	0.342***	-0.028	-5.743**	5.715***
former us treasury secretary	806	38	0.112	-1.282	-9.506***	8.224***
chinese president	482	95	0.357***	1.106	-4.832**	5.938***
half point	997	58	0.276***	-0.544	-6.268***	5.724***
chinese president xi	568	93	0.352***	1.155	-4.703**	5.858***
say country	963	98	0.164**	-0.361	-5.856***	5.496***
past year	198	197	0.247***	1.261	-3.255	4.515***
russia invasion ukraine	115	216	0.520***	-0.743	-5.600***	4.857***
secretary janet	168	126	0.374***	0.930	-4.789**	5.719***
secretary janet yellen	169	126	0.374***	0.930	-4.789**	5.719***
justice department	232	131	0.157*	-0.426	-5.815**	5.389***
former us treasury	805	41	0.108	-1.261	-8.684***	7.423***
young people	699	99	0.231***	0.187	-5.466**	5.653**
economist say	973	86	0.064	1.383	-4.132*	5.515**
local government	779	87	0.116*	0.052	-5.044**	5.096**
elon musk say	540	78	0.223*	-0.906	-6.070**	5.164**
emerge economic	787	21	0.859***	1.687	-4.427	6.114**
emerge economic powerhouse	808	21	0.850***	1.687	-4.427	6.114**
government plan	775	110	0.036	-0.108	-4.272**	4.164**
new book	901	91	0.156*	0.425	-4.347*	4.773**
come next	837	79	0.089	-2.223**	-7.530***	5.307**
asylum seeker	864	68	0.297***	-1.033	-5.792**	4.759**
next big	487	96	0.319***	-3.119***	-8.000***	4.881**

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 19. Persistence of ChatGPT Day Trading Signals

This table reports the panel regression results of examining the persistence of ChatGPT day trading signals. Columns (1) to (4) use *BUY* as the main dependent variable. Columns (5) to (8) use *SELL* as the main dependent variable. *BUY* (*SELL*) is a dummy variable having 1 when a stock is selected by ChatGPT to buy (sell). $BUY\ (t-1)$ is the previous day's *BUY* signal, and $SELL\ (t-1)$ is the previous day's *SELL* signal. All continuous independent variables are winsorized at 1% and 99% level. All estimations include date-fixed effects. The standard errors are two-way clustered by stock- and day-level, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	BUY				*SELL*			
$BUY\ (t-1)$	0.533***		0.410***	0.317***	0.505***		0.347***	0.270***
	(52.54)		(53.74)	(48.23)	(57.20)		(62.42)	(52.01)
$SELL\ (t-1)$		0.410***	0.235***	0.183***		0.449***	0.301***	0.257***
		(43.01)	(54.76)	(48.62)		(53.09)	(57.83)	(50.69)
SIZE				0.037***				0.030***
				(38.92)				(29.69)
TURN				0.999***				1.324***
				(15.14)				(16.96)
D_TWTR				0.180***				0.147***
				(20.13)				(18.40)
Observations	1059956	1059956	1059956	1059956	1059956	1059956	1059956	1059956
R-sq	0.289	0.212	0.341	0.386	0.214	0.209	0.284	0.310

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 20. Persistence of *NET_BUY* and *NET_SELL* Signals

This table reports the panel regression results of regressing *NET_BUY* and *NET_SELL* signals on their lagged signals. Columns (1) use the *NET_BUY* signal as the main dependent variable. Column (2) uses the *NET_SELL* signal as the main dependent variable. *NET_BUY* $(t-1)$ is the one-day lagged *NET_BUY* signal, and *NET_SELL* $(t-1)$ is the one-day lagged *NET_SELL* signal. *D_NET_SELL* is a dummy variable having 1 when the *NET_SELL* signal is non-zero. All continuous independent variables are winsorized at 1% and 99% level. All estimations include date-fixed effects. The standard errors are two-way clustered by stock- and day-level, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

	(1) *NET_BUY* (t)	(2) *NET_SELL* (t)
NET_BUY $(t-1)$	0.738***	-0.015***
	(27.07)	(-13.29)
NET_SELL $(t-1)$	0.012	0.532***
	(1.44)	(17.32)
D_NET_SELL $(t-1)$	0.074***	-0.311***
	(5.53)	(-9.59)
SIZE	0.075***	0.038***
	(11.80)	(17.27)
TURN	0.864***	2.787***
	(3.09)	(10.09)
D_TWTR	1.995***	0.050
	(8.68)	(1.64)
Observations	1059662	1059662
R-sq	0.629	0.232

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 21. Persistence of Intraday and Overnight *NET_BUY* and *NET_SELL* Signals

This table reports the panel regression results of regressing *Intraday* and *Overnight News*-based *NET_BUY* and *NET_SELL* signals on lagged values. *Intraday News* $(t-1)$ and *Overnight News* $(t-1)$. *Intraday News* $(t-1)$ indicates tweets posted from 9:30 AM to 4:00 PM of day $t-1$. *Overnight News* $(t-1)$ indicates tweets posted from 4:00 PM of day $t-1$ to the next trading day's market opening at 9:30 AM. *Intraday News* (t) and *Overnight News* (t) are similarly defined. Columns (1) and (2) use the *NET_BUY* and *NET_SELL* signals generated by using *Intraday News* (t) as the main dependent variables. Columns (3) and (4) use the *NET_BUY* and *NET_SELL* signals generated by *Overnight News* (t) as the main dependent variable. *D_NET_SELL* is a dummy variable having 1 when the *NET_SELL* signal is non-zero. Stocks with zero *NET_BUY* or *NET_SELL* signals are included in the regressions. All continuous independent variables are winsorized at 1% and 99% level. All estimations include date-fixed effects. The standard errors are two-way clustered by stock- and day-level, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

	(1)	(2)	(3)	(4)
News	*Intraday News* (t)		*Overnight News* (t)	
	NET_BUY	*NET_SELL*	*NET_BUY*	*NET_SELL*
NET_BUY by *Overnight News* $(t-1)$	0.941***	-0.011***		
	(24.06)	(-4.33)		
NET_SELL by *Overnight News* $(t-1)$	0.066***	0.617***		
	(3.50)	(13.13)		
D_NET_SELL by *Overnight News* $(t-1)$	0.076***	-0.417***		
	(3.94)	(-9.46)		
NET_BUY by *Intraday News* (t)			0.448***	-0.007***
			(21.65)	(-5.92)
NET_SELL by *Intraday News* (t)			0.044***	0.302***
			(4.30)	(14.73)
D_NET_SELL by *Intraday News* $(t-1)$			0.027**	-0.116***
			(2.55)	(-5.92)
SIZE	0.040***	0.028***	0.059***	0.025***
	(11.80)	(16.33)	(14.05)	(20.97)
TURN	0.379**	2.040***	0.779***	1.695***
	(2.01)	(9.57)	(4.41)	(10.28)
D_TWTR	1.093***	0.037	1.184***	0.001
	(7.21)	(1.41)	(9.06)	(0.06)
Observations	1059956	1059956	1059662	1059662
R-sq	0.516	0.151	0.519	0.135

Electronic copy available at: https://ssrn.com/abstract=4759311

Table 22. Effects of Limits-of-Arbitrage

This table reports the portfolio performance conditional on limits-of-arbitrage. All portfolios are constructed based on *NET_BUY* and *NET_SELL* signals. Portfolio performance is intraday CAPM alpha that is reported in monthly percentage terms. Panel A uses firm size as a proxy of limits-of-arbitrage. Firm size is measured by the previous trading day's market capitalization. In each of the stocks with positive *NET_BUY* or *NET_SELL* signals, I divide stocks into *Small* and *Large* firms based on their size. Panel B uses the average of rank-transformed variables proxying trading frictions, including share turnover, illiquidity, bid-ask spread, and return volatility. In each of the stocks with positive *NET_BUY* or *NET_SELL* signals, I divide stocks into *High* and *Low* groups based on their average rank of trading frictions. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. Firm Size

Small			*Large*		
NET_BUY	*NET _SELL*	*NET _BMS*	*NET _BUY*	*NET _SELL*	*NET _BMS*
-1.176	-5.461***	4.285***	0.835**	-1.401*	2.236**
(-0.91)	(-2.90)	(3.50)	(2.34)	(-1.66)	(2.48)

Panel B. Trading Frictions

High			*Low*		
NET_BUY	*NET _SELL*	*NET _BMS*	*NET _BUY*	*NET _SELL*	*NET _BMS*
-0.189	-5.212***	5.023***	0.757**	-1.410*	2.167***
(-0.16)	(-2.70)	(3.53)	(2.57)	(-1.85)	(2.71)

Electronic copy available at: https://ssrn.com/abstract=4759311

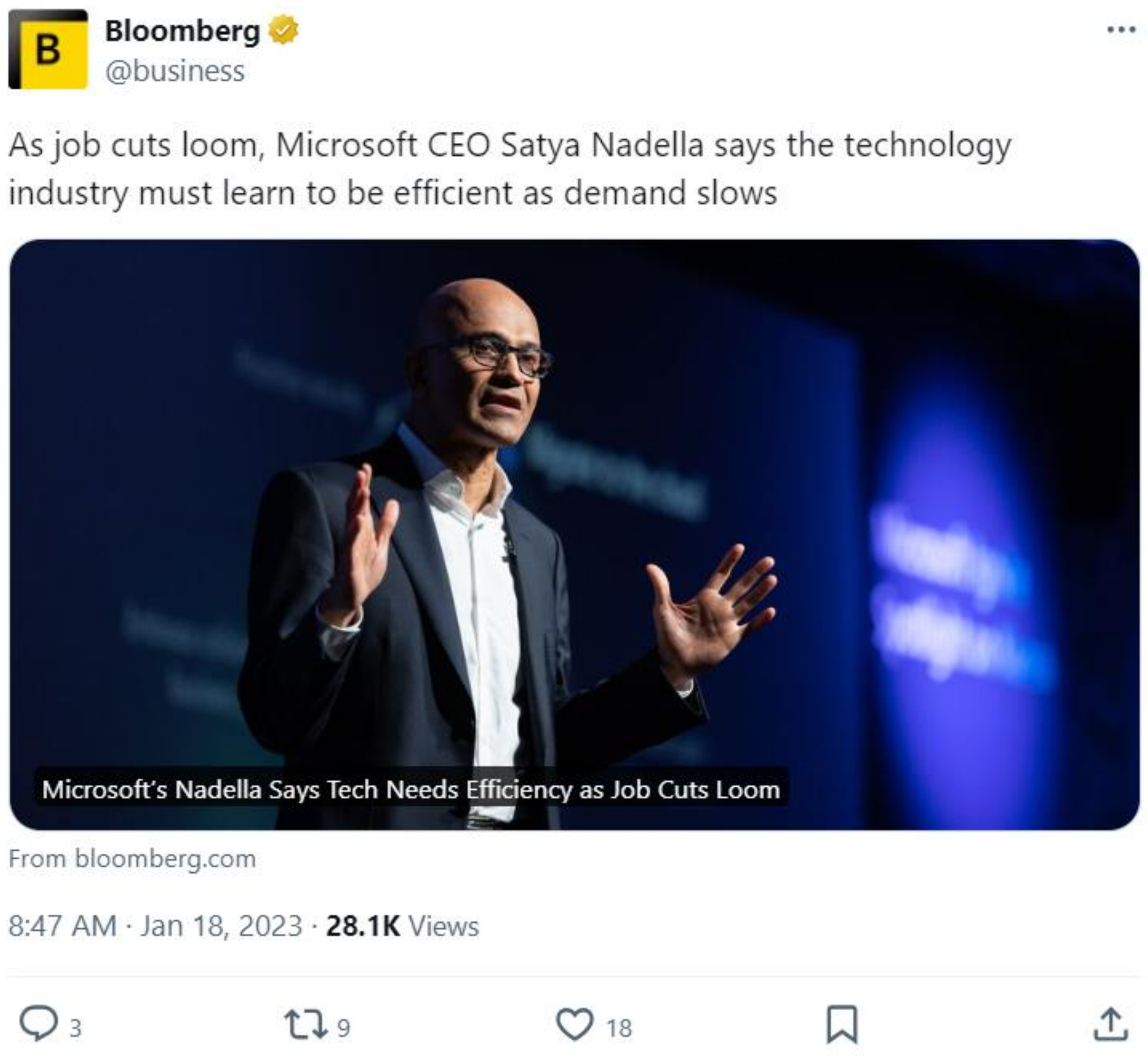
Appendix

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Figure 1. Firm-Specific News Examples

This figure shows Twitter pages and the corresponding Bloomberg pages for the first and second examples in Table 1. Panel A shows the Twitter page related to Microsoft (Ticker: MSFT), the first example in Table 1. Panel B shows the corresponding Bloomberg news page for Panel A. Panel C shows the Twitter page related to Bed Bath & Beyond (BBBY), the second example in Table 1. Panel D shows the corresponding Bloomberg news page for Panel C.

Panel A. Twitter Page for the First Example in Table 1, which is about Microsoft (Ticker: MSFT)



Bloomberg
@business

As job cuts loom, Microsoft CEO Satya Nadella says the technology industry must learn to be efficient as demand slows

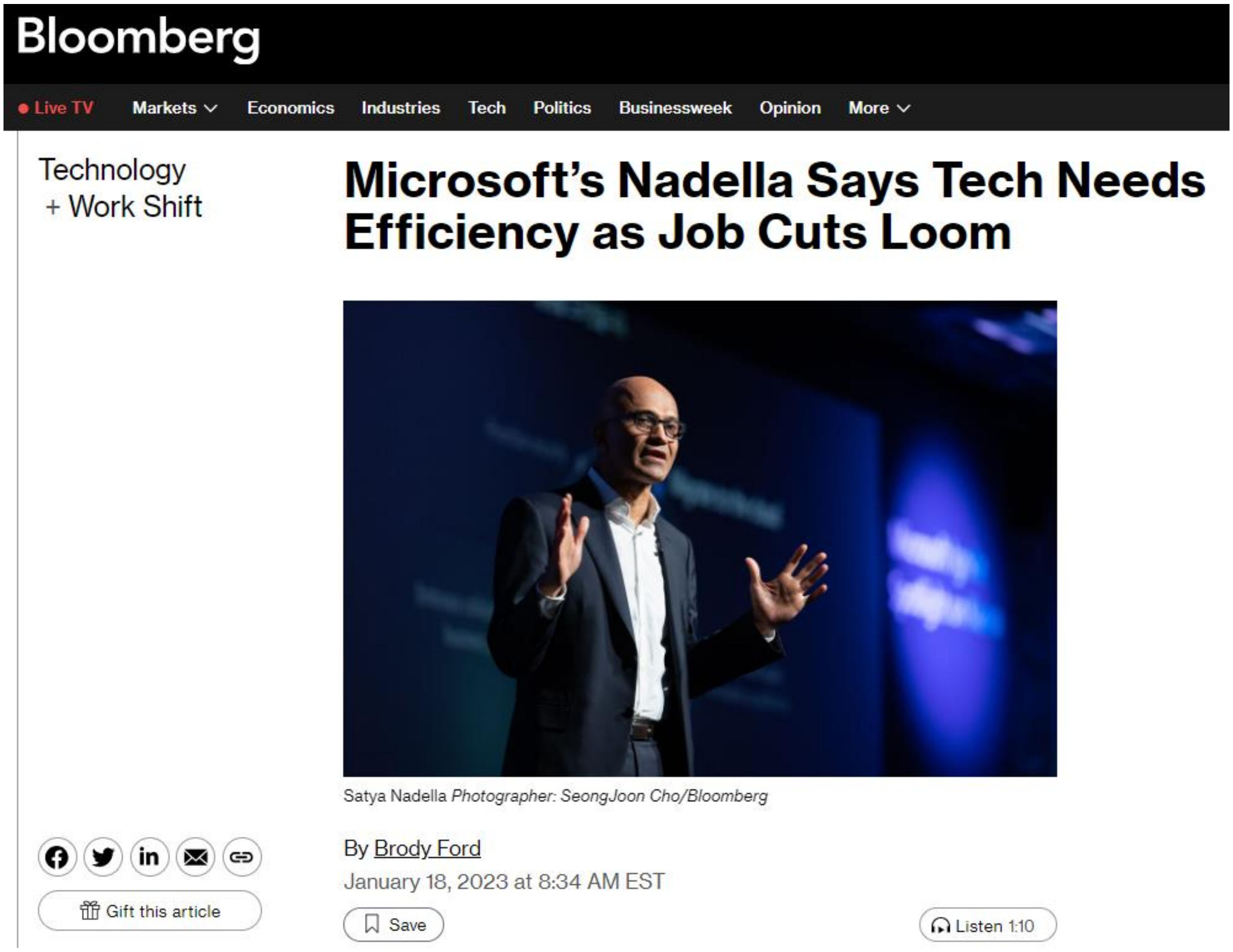
Microsoft's Nadella Says Tech Needs Efficiency as Job Cuts Loom

From bloomberg.com

8:47 AM · Jan 18, 2023 · **28.1K** Views

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel B. Bloomberg Page for Panel A



Microsoft’s Nadella Says Tech Needs Efficiency as Job Cuts Loom

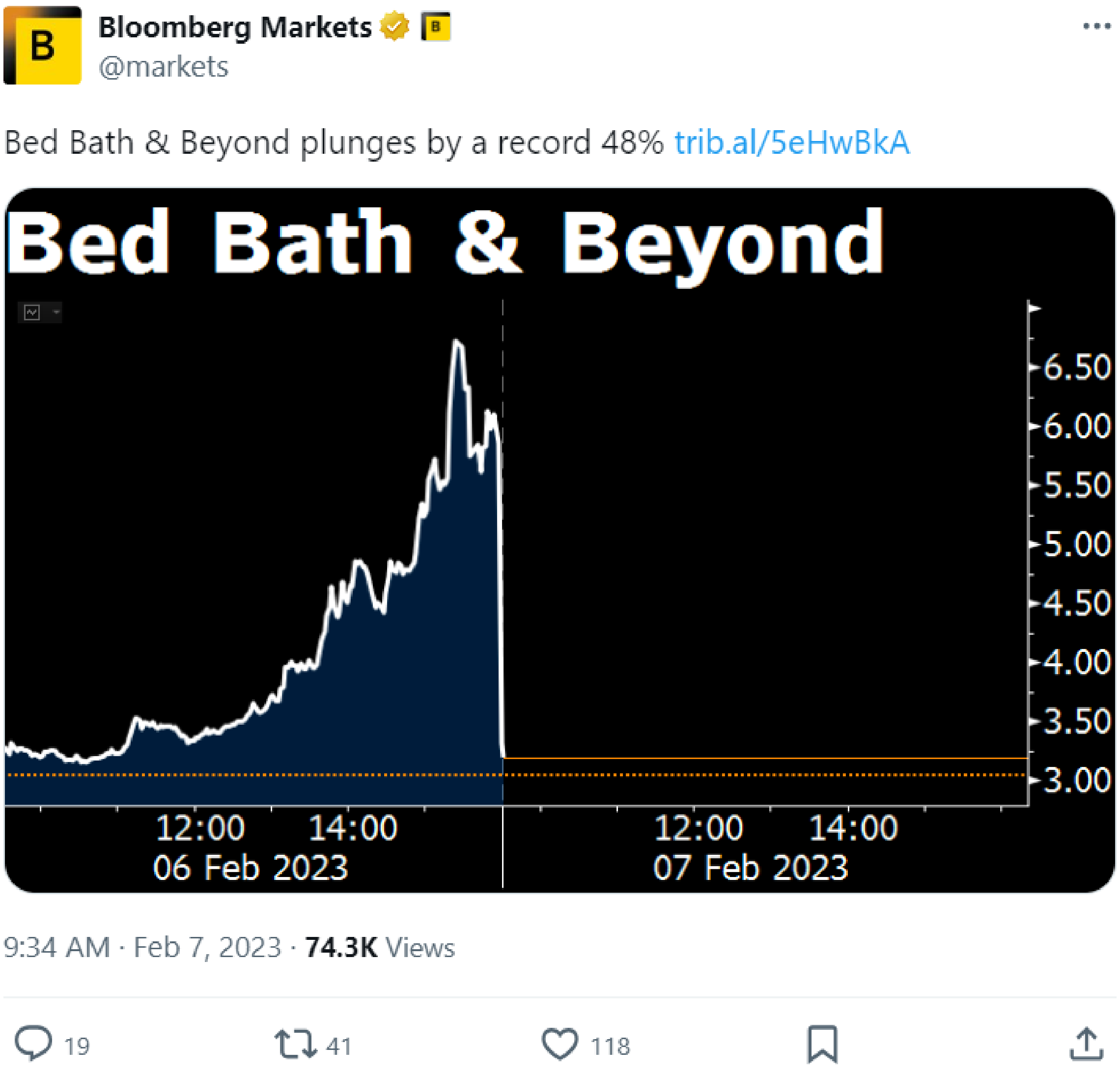
Satya Nadella *Photographer: SeongJoon Cho/Bloomberg*

By Brody Ford

January 18, 2023 at 8:34 AM EST

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel C. Twitter Page for the Second Example in Table 1, which is about Bed Bath & Beyond (Ticker: BBBY)



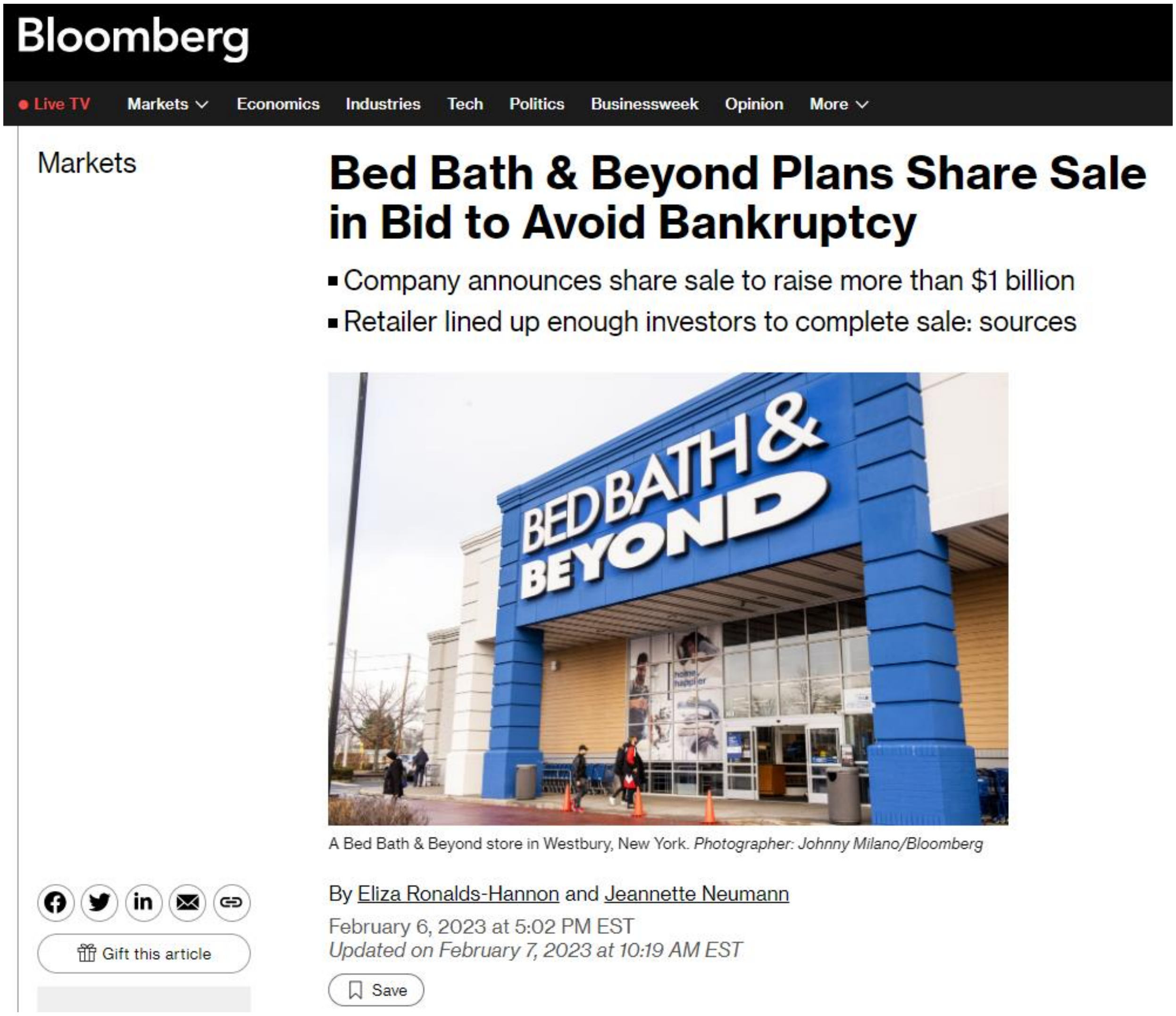
Bloomberg Markets
@markets

Bed Bath & Beyond plunges by a record 48% trib.al/5eHwBkA

9:34 AM · Feb 7, 2023 · **74.3K** Views

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel D. Bloomberg Page for Panel C



Bed Bath & Beyond Plans Share Sale in Bid to Avoid Bankruptcy

- Company announces share sale to raise more than $1 billion
- Retailer lined up enough investors to complete sale: sources

A Bed Bath & Beyond store in Westbury, New York. *Photographer: Johnny Milano/Bloomberg*

By Eliza Ronalds-Hannon and Jeannette Neumann

February 6, 2023 at 5:02 PM EST

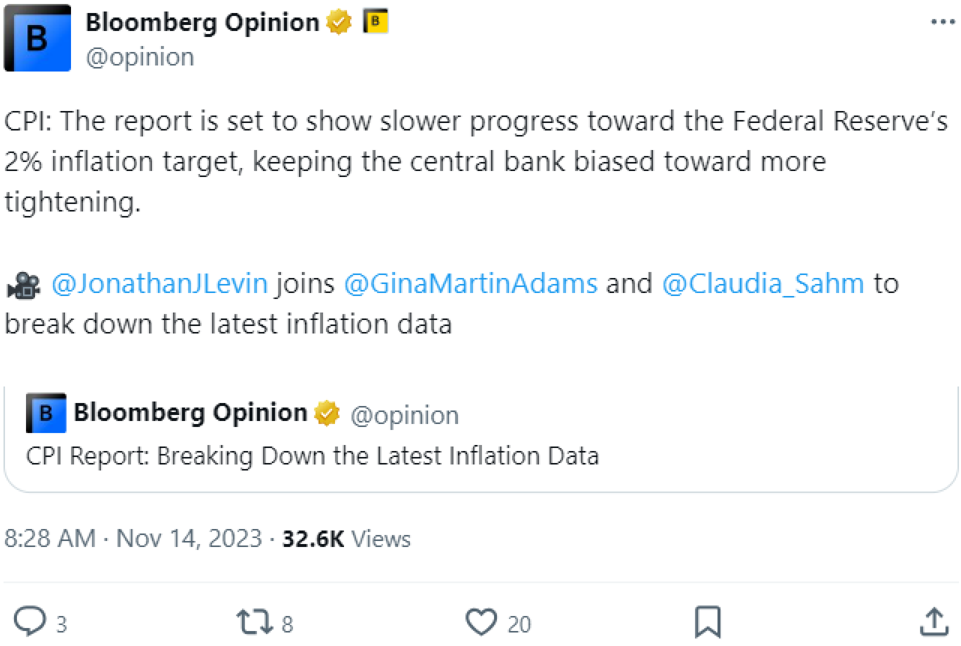
Updated on February 7, 2023 at 10:19 AM EST

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Figure 2. Macro News Examples

This figure shows Twitter pages and the corresponding Bloomberg pages for the third and the remaining examples in Table 1. Panels A, B, and C show the Twitter page related to CPI releases, which are the third, fourth, and fifth examples in Table 1. Panel A is for the before CPI releases and Panel B is for the after CPI releases. Panel C is for the energy prices. Panel D shows the Twitter page about the Job Openings and Labor Turnover Survey (JOLTS) Program, which is about the last example in Table 1. Panel E shows the corresponding Bloomberg page for Panel D.

Panel A. Before CPI Release



Bloomberg Opinion
@opinion

CPI: The report is set to show slower progress toward the Federal Reserve's 2% inflation target, keeping the central bank biased toward more tightening.

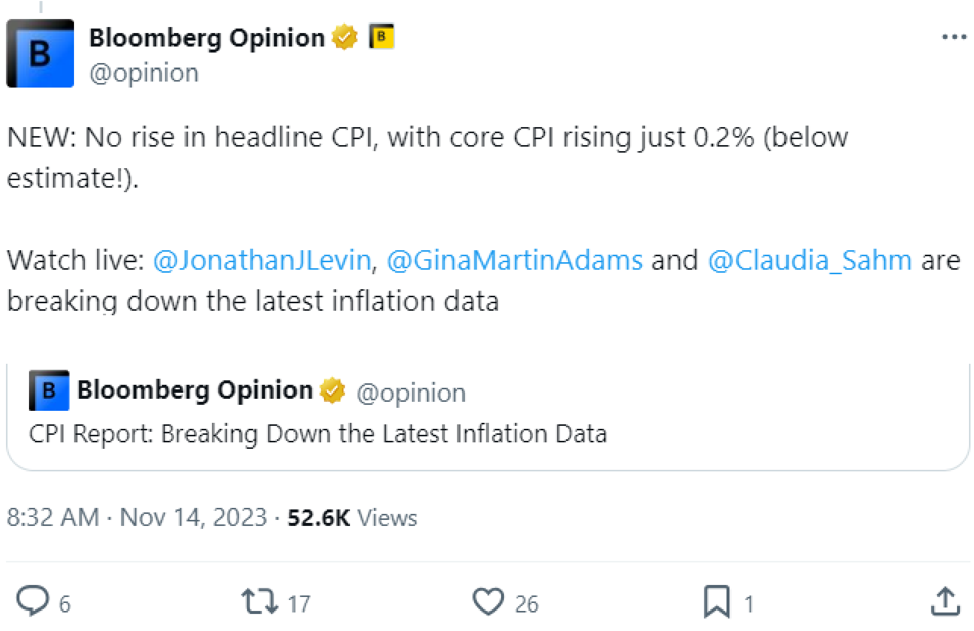
🎥 @JonathanJLevin joins @GinaMartinAdams and @Claudia_Sahm to break down the latest inflation data

Bloomberg Opinion @opinion
CPI Report: Breaking Down the Latest Inflation Data

8:28 AM · Nov 14, 2023 · **32.6K** Views

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel B. After CPI Release



Bloomberg Opinion
@opinion

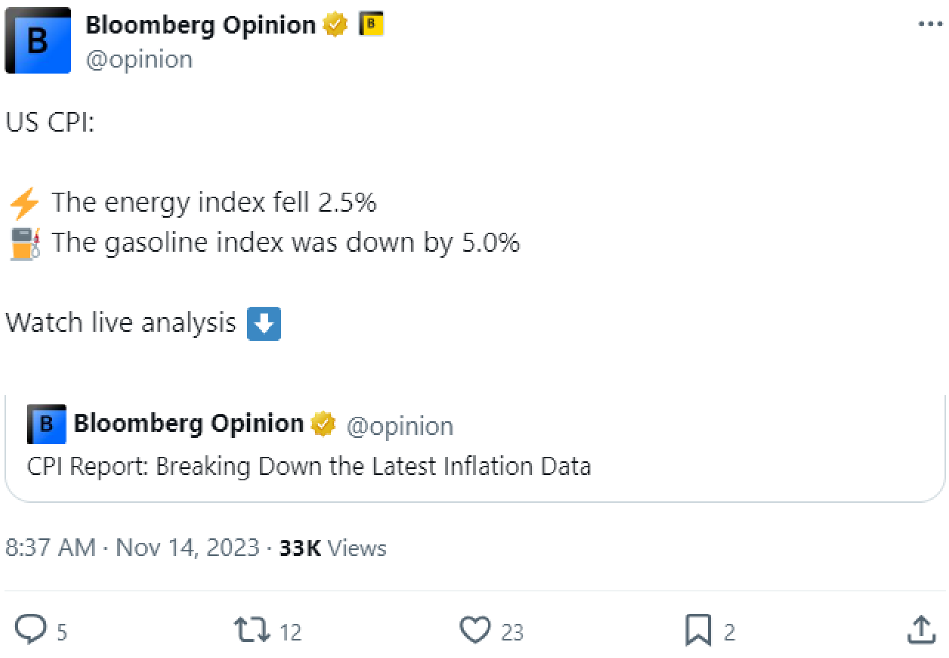
NEW: No rise in headline CPI, with core CPI rising just 0.2% (below estimate!).

Watch live: @JonathanJLevin, @GinaMartinAdams and @Claudia_Sahm are breaking down the latest inflation data

Bloomberg Opinion @opinion
CPI Report: Breaking Down the Latest Inflation Data

8:32 AM · Nov 14, 2023 · **52.6K** Views

Panel C. Energy Prices



Bloomberg Opinion
@opinion

US CPI:

⚡ The energy index fell 2.5%
⛽ The gasoline index was down by 5.0%

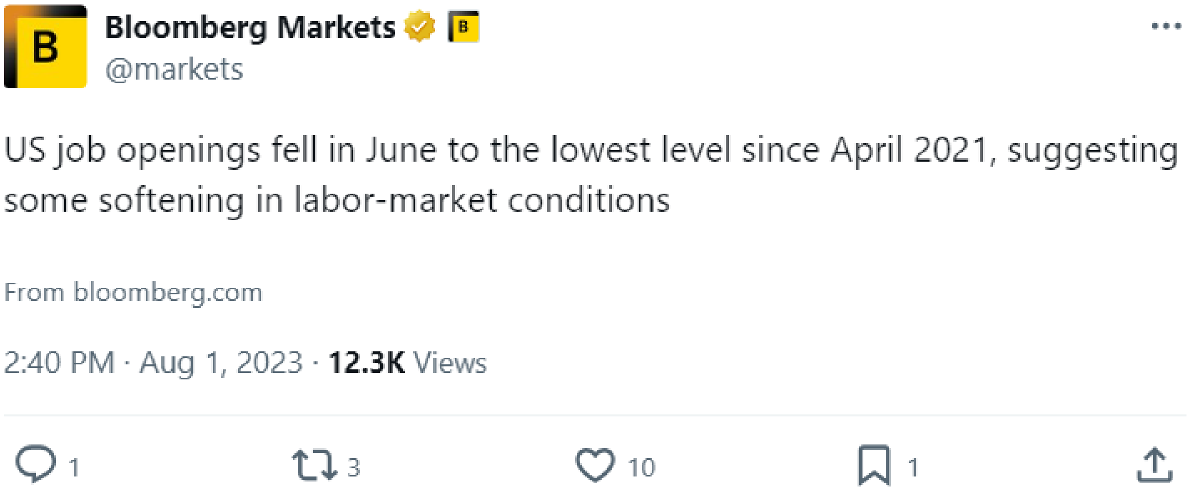
Watch live analysis ⬇️

Bloomberg Opinion @opinion
CPI Report: Breaking Down the Latest Inflation Data

8:37 AM · Nov 14, 2023 · **33K** Views

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel D. Twitter Page about the Job Openings and Labor Turnover Survey (JOLTS) Program

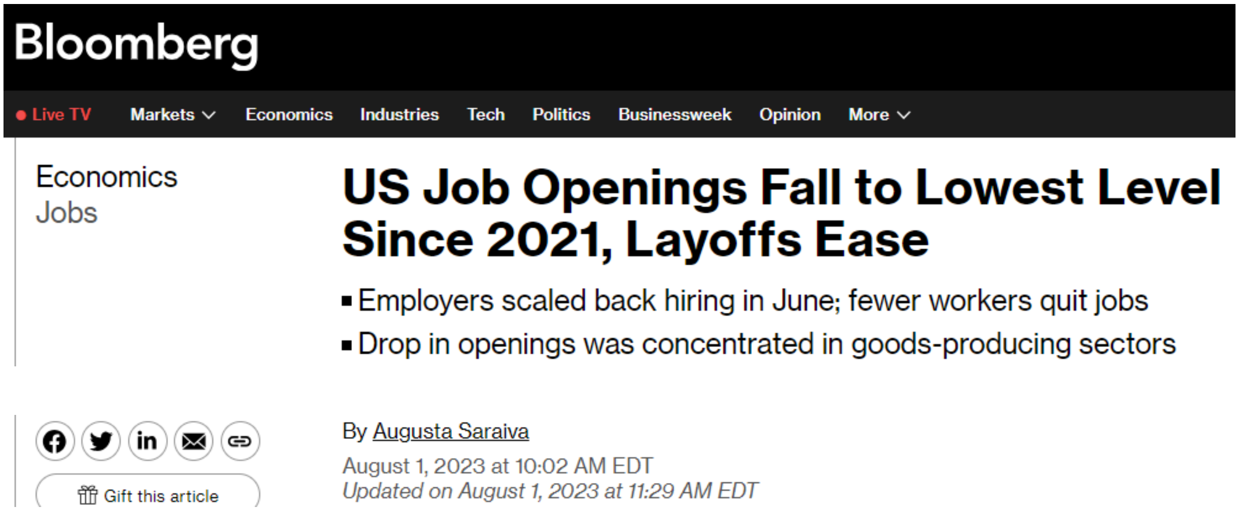


US job openings fell in June to the lowest level since April 2021, suggesting some softening in labor-market conditions

From bloomberg.com

2:40 PM · Aug 1, 2023 · **12.3K** Views

Panel E. Bloomberg Page for Panel D



US Job Openings Fall to Lowest Level Since 2021, Layoffs Ease

- Employers scaled back hiring in June; fewer workers quit jobs
- Drop in openings was concentrated in goods-producing sectors

By Augusta Saraiva
August 1, 2023 at 10:02 AM EDT
Updated on August 1, 2023 at 11:29 AM EDT

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Table 1. Stock-Level Characteristics and the ChatGPT Day Trading Signals with Industry Fixed Effects

This table reports panel regression results of testing determinants of the ChatGPT signals with day-industry fixed effects. The industry classification is CRSP SIC two-digit codes (SIC2). All estimations include date-SIC2-interacted fixed effects. The main dependent variables are *BUY* and *SELL* signals. *BUY* (*SELL*) signal is a dummy variable having 1 when a stock is selected at least once as a stock buy (sell). Panel A uses stock characteristics as the main independent variables. All independent variables are lagged by one day. *SIZE* is the log of market capitalization. *MOM1M* is the previous 20 days of returns. *MOM12M* is the previous 12 to 1-month returns. *RET* $(t-1)$ is the previous day's returns. Panel B uses an indicator for Twitter mentioned and various market information as the main independent variables. *D_TWTR* is a dummy variable having 1 one when a stock apperas in a tweet, which is classified by Refinitiv Eikon News Monitor. *N_TWTR* is the total number of tweets that a stock appears. *D_SP500* is a dummy variable having 1 one a stock is a constituent of the S&P 500 index. *D_NASDAQ* is a dummy variable having 1 one a stock is traded in NASDAQ. Panel C uses trading frictions as the main independent variables. All independent variables are lagged by one day. *TURN* is the previous trading day's share turnover. *ILLIQ* is the Amihud (2002) illiquidity measure multiplied by 10,000. *BASPREAD* is the bid-ask spread, and *RETVOL* is the previous 20 days of return volatility. Panel D uses earnings announcement day as the main independent variable. *D_RDQ* is a dummy variable having 1 when a firm announced earnings in the previous 7 days. All continuous independent variables are winsorized at 1% and 99% level. The standard errors are two-way clustered by stock- and day-level, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel A. Stock-Level Characteristics

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	BUY					*SELL*				
SIZE	0.073***				0.085***	0.064***				0.077***
	(39.57)				(43.86)	(35.90)				(41.09)
MOM1M		0.045***			-0.112***		0.027***			-0.113***
		(5.53)			(-16.26)		(3.38)			(-14.08)
MOM12M			0.030***		-0.138***			0.011*		-0.139***
			(4.82)		(-21.73)			(1.72)		(-20.24)
$RET\ (t-1)$				0.079***	-0.017				0.028	-0.043*
				(3.00)	(-0.80)				(1.05)	(-1.94)
Observations	1058198	1058198	1032451	1058198	1032451	1058198	1058198	1032451	1058198	1032451
R-sq	0.260	0.078	0.078	0.077	0.284	0.197	0.081	0.080	0.081	0.216

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel B. Tweets and Market Information

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	BUY					*SELL*				
D_TWTR	0.593***				0.196***	0.546***				0.188***
	(25.00)				(13.27)	(22.98)				(13.04)
N_TWTR		0.093***			-0.003		0.085***			-0.001
		(9.02)			(-1.51)		(9.26)			(-0.56)
D_SP500			0.609***		0.436***			0.548***		0.398***
			(47.03)		(30.13)			(45.58)		(28.35)
D_NASDAQ				-0.082***	0.011				-0.102***	-0.021**
				(-6.46)	(1.41)				(-8.35)	(-2.51)
SIZE					0.041***					0.034***
					(22.24)					(17.98)
Observations	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198
R-sq	0.095	0.086	0.303	0.084	0.346	0.093	0.087	0.230	0.089	0.256

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel C. Trading Frictions

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	BUY					*SELL*				
TURN	1.283***				1.690***	1.714***				2.066***
	(7.20)				(12.02)	(9.27)				(13.42)
ILLIQ		-5.416***			5.347***		-5.370***			4.205***
		(-19.27)			(19.58)		(-18.16)			(14.65)
BASPREAD			-1.132***		0.207***			-0.918***		0.135***
			(-19.03)		(5.97)			(-15.25)		(3.55)
RETVOL				-1.675***	0.747***				-1.301***	0.726***
				(-16.53)	(12.19)				(-12.62)	(10.47)
SIZE					0.087***					0.077***
					(41.44)					(37.37)
Observations	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198
R-sq	0.081	0.088	0.091	0.089	0.280	0.087	0.090	0.088	0.087	0.213

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel D. Earnings Announcement

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	BUY					*SELL*				
D_RDQ	0.007	0.005	0.002	-0.000	-0.009***	0.003	0.001	-0.001	-0.007	-0.015***
	(1.02)	(1.29)	(0.28)	(-0.04)	(-2.62)	(0.60)	(0.36)	(-0.26)	(-1.12)	(-4.15)
SIZE		0.073***			0.073***		0.065***			0.065***
		(39.04)			(39.78)		(35.44)			(36.64)
D_TWTR			0.593***		0.315***			0.546***		0.295***
			(24.98)		(18.80)			(22.98)		(16.75)
TURN				1.284***	1.960***				1.719***	2.318***
				(7.17)	(14.99)				(9.25)	(16.23)
Observations	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198	1058198
R-sq	0.077	0.258	0.095	0.081	0.273	0.081	0.196	0.093	0.087	0.211

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Table 2. Top 31 to 90 Important Keywords by P-Values, Their Persistentcy, and Impacts on the Day Trading Strategy by ChatGPT Signals

This table reports the top 31 to 90 significant keywords for the analysis in Table 18.

Panel A. Top 31 to 60 Significance				$\beta_{Keyword}$		
Keyword	*Rank*	*N*	*AR Coef*	*NET_BUY*	*NET_SELL*	*NET_BMS*
survey show	692	126	0.357***	0.161	-4.387**	4.548**
former us	251	95	0.136***	-1.420	-6.171***	4.751**
treasury secretary	78	154	0.170**	1.333*	-3.374	4.707**
chancellor jeremy	254	89	0.306**	0.609	-4.383*	4.991**
chancellor jeremy hunt	255	89	0.306**	0.609	-4.383*	4.991**
fastest growing	933	78	0.226***	-0.921	-5.904**	4.983**
president erdogan	796	94	0.130*	-0.820	-5.341**	4.521**
past month	695	103	0.114	0.475	-3.997*	4.473**
jeremy hunt	130	116	0.381***	1.218	-3.122	4.341**
new survey	708	109	0.303**	0.232	-4.266*	4.498**
dollar bond	861	96	0.208**	-1.002	-5.245**	4.243**
oil production	505	93	0.392***	-0.535	-4.677**	4.142**
us treasury secretary	319	86	0.231***	0.389	-4.302*	4.691**
watch full	145	134	0.332***	-1.143	-5.560***	4.417**
us secretary	899	67	0.402***	-1.011	-5.799**	4.788**
getting warmer	790	48	0.545***	-1.409	-7.235**	5.826**
collapse silicon	992	27	0.755***	-1.296	-8.091**	6.795**
collapse silicon valley	993	27	0.755***	-1.296	-8.091**	6.795**
musk say	406	90	0.278**	-0.168	-4.433*	4.264**
us government	124	182	0.173	0.611	-3.346	3.956**
live update	954	65	0.218	-1.835**	-6.871**	5.036**
uk economy	264	130	0.141**	0.481	-3.370	3.851**
far right	703	78	0.308***	-1.659*	-5.837**	4.178**
authority say	745	100	0.311***	0.017	-3.641*	3.657**
investment firm	508	123	0.222***	-0.134	-3.659*	3.526**
file bankruptcy	497	97	0.098	-0.289	-3.925*	3.636**
cold war	942	68	0.178**	0.980	-3.823	4.803*
us dollar	330	131	0.229***	0.115	-2.993	3.108*
green energy	810	104	0.067	-0.404	-4.815*	4.411*
attorney general	605	113	0.053	1.587*	-1.990	3.578*

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel B. Top 61 to 90 Significance

				$\beta_{Keyword}$		
Keyword	*Rank*	*N*	*AR Coef*	*NET_BUY*	*NET_SELL*	*NET_BMS*
new episode	363	80	0.491***	-1.874*	-6.168**	4.294*
india edition newsletter	222	46	0.835***	0.303	-3.617	3.920*
economic outlook	627	108	0.098**	-0.055	-3.600*	3.545*
presidential candidate	154	102	0.514***	-1.488*	-5.245**	3.757*
say need	947	95	0.159*	-1.036	-4.792*	3.757*
volodymyr zelenskiy	265	89	0.317***	-0.880	-4.592*	3.712*
make sense	667	101	0.222***	-1.287	-4.724**	3.437*
greenhouse gas	725	91	0.019	-0.346	-4.093*	3.747*
credit card	276	137	0.088	1.362	-2.536	3.897*
us economy	109	192	0.187**	0.862	-2.760	3.622*
coming month	647	116	0.084	-2.230**	-5.829**	3.599*
signature bank	768	47	0.552***	-0.877	-6.162*	5.285*
rate high	616	113	0.078*	-1.733*	-5.618**	3.885*
policy maker	639	103	0.185**	0.335	-3.499	3.834*
global economic	988	108	0.141	-0.451	-4.044*	3.593*
support ukraine	814	92	0.240***	-0.566	-4.522*	3.956*
us lawmaker	803	67	0.361***	-0.566	-4.488*	3.922*
bank failure	476	53	0.584***	-0.620	-4.819*	4.199*
street week	208	62	0.082	-0.131	-4.078*	3.947*
bank raise	452	106	0.102	-1.822**	-5.098**	3.276*
say us	122	207	0.069	0.500	-2.537	3.038*
billion dollar	104	220	0.123***	0.025	-3.523	3.549*
first time month	747	98	-0.011	-0.611	-3.863*	3.252*
cyril ramaphosa	855	66	0.463***	0.554	-2.811	3.365*
pave way	459	116	0.097*	1.545*	-1.801	3.346*
invest right	681	95	0.249***	-1.136	-4.533**	3.397*
venture capital	610	112	0.174***	-0.797	-4.627*	3.830*
recent month	902	99	0.052	0.031	-3.726	3.758*
us city	509	95	0.310***	-0.141	-3.831	3.689*
million dollar	370	131	0.114*	-1.488*	-4.820**	3.332*

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Table 3. Release of Macroeconomic Indicators

This table reports the results of examining the effects of the release of macroeconomic indicators on portfolio performance constructed by ChatGPT day trading signals. Panel A reports summary statistics of the releases of macroeconomic indicators. Macroeconomic indicators are collected from the Bloomberg Economic Calendars (ECO). The macroeconomic indicator has 13 categories, which are reported in Panel A. The column labeled Mean reports the proportion of days, with a macroeconomic indicator category specified in each row. #Days Released shows the number of days for each macroeconomic indicator category. #Indicators per Day shows the average number of macroeconomic indicators released per day for each macroeconomic indicator category. Max #Indicators shows the maximum number of macroeconomic indicators released per day for each macroeconomic indicators category. Panel B shows the time-series regression results of regressing *NET_BMS* portfolio returns. Specifically, Panel B estimates the following time-series regression:

$$NET\_BMS_t = Constant + \beta_{Market} \cdot MKTRF_t + \sum_i \beta_i \cdot Macro\ Indicator_{i,t-window\ to\ t-lag} + \epsilon_t.$$

The *lag* can be 0 or 1, which is denoted in the row labeled *Lag*, which has 0 for contemporaneous macroeconomic indicator releases and 1 for lagged releases. The window can be 1 to 5, which is denoted by the row labeled as *Window*. $Macro\ Indicator_{i,t-window\ to\ t-lag}$ is 1 when at least one macroeconomic indicator of category i is released during $t - window$ and $t - lag$. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. Summary Statistics

Macro Indicator	Mean	#Days Released	#Indicators per Day	Max #Indicators
Government Finance, Debt	4.80%	13	1.00	1
Housing and Real Estate	37.64%	102	2.32	7
Industrial Sector	23.25%	63	1.84	3
International Trade, BoP	15.87%	43	2.21	5
Labor Market	34.69%	94	3.23	11
Monetary Sector	3.32%	9	2.00	2
National Accounts (GDP)	4.80%	13	2.00	2
Personal, Household Sector	9.59%	26	3.69	6
Prices	29.15%	79	3.51	8
Purchasing Managers' Index	19.93%	54	3.31	6
Retail, Wholesale Sector	14.02%	38	2.34	4
Services Sector	9.59%	26	1.00	1
Surveys, Cyclical Indicators	60.89%	165	2.81	7

Electronic copy available at: https://ssrn.com/abstract=4759311

Panel B. Regression Result

	(1)	(2)	(3)	(4)	(5)	(6)
Lag	0	1	1	1	1	1
Size of Moving Window	1	1	2	3	4	5
	NET_BMS Return (monthly, %)					
Government Finance, Debt	0.546	-9.376*	-6.973	-3.069	-2.204	-2.613
	(0.11)	(-1.75)	(-1.36)	(-0.81)	(-0.67)	(-0.85)
Housing and Real Estate	-0.067	-3.507*	-4.880**	-4.344	4.304	
	(-0.03)	(-1.78)	(-2.05)	(-1.41)	(0.74)	
Industrial Sector	0.013	-1.765	-1.709	-2.094	-2.537	-1.849
	(0.00)	(-0.53)	(-0.66)	(-0.88)	(-0.92)	(-0.67)
International Trade, BoP	-1.990	2.210	3.397	3.031	1.012	4.465
	(-0.53)	(0.62)	(1.55)	(1.28)	(0.41)	(1.57)
Labor Market	-0.171	-1.364	0.970	1.657	1.320	
	(-0.08)	(-0.66)	(0.42)	(0.72)	(0.49)	
Monetary Sector	-0.845	-7.349	-7.185**	-2.316	-1.548	0.119
	(-0.12)	(-1.31)	(-2.40)	(-0.79)	(-0.54)	(0.04)
National Accounts (GDP)	2.644	-1.522	1.258	-1.713	-1.981	-0.458
	(0.66)	(-0.31)	(0.33)	(-0.49)	(-0.61)	(-0.13)
Personal, Household Sector	1.434	1.995	-1.805	0.469	1.535	-0.922
	(0.34)	(0.61)	(-0.64)	(0.16)	(0.51)	(-0.34)
Prices	-0.962	0.880	1.317	0.859	1.073	0.500
	(-0.28)	(0.31)	(0.50)	(0.32)	(0.38)	(0.17)
Purchasing Managers' Index	-2.522	-1.965	-1.222	-0.295	-2.666	-0.744
	(-0.70)	(-0.42)	(-0.30)	(-0.08)	(-0.67)	(-0.18)
Retail, Wholesale Sector	-1.760	-1.970	-1.805	-1.674	-0.153	2.921
	(-0.50)	(-0.64)	(-0.74)	(-0.69)	(-0.06)	(1.21)
Services Sector	2.615	5.346	0.859	1.401	3.872	0.524
	(0.63)	(0.90)	(0.17)	(0.30)	(0.92)	(0.14)
Surveys, Cyclical Indicators	3.121	-5.212**	-6.813**	-8.893*	-1.939	-5.257
	(1.14)	(-2.08)	(-2.21)	(-1.93)	(-0.32)	(-1.14)
SPY	-0.191***	-0.194***	-0.195***	-0.186***	-0.198***	-0.198***
	(-3.23)	(-3.57)	(-3.32)	(-3.09)	(-3.40)	(-3.45)
Constant	2.730	9.218***	13.239***	14.805**	1.313	6.158
	(0.93)	(3.73)	(3.61)	(2.29)	(0.15)	(1.03)

Electronic copy available at: https://ssrn.com/abstract=4759311

Observations	271	271	271	271	271	271
R-sq	-0.009	0.035	0.039	0.021	-0.006	0.009

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Table 4. Fama-Macbeth Regression

This table reports the Fama-Macbeth regression results of regressing intraday excess returns on ChatGPT day trading signals. The main dependent variable is intraday excess return, which is open-to-close price returns in basis point terms. The main independent variables are *NET_SIG*, *NET_BUY* and *NET_SELL* signals. *NET_SIG* is the the difference between *SUM_BUY* and *SUM_SELL*. There are various control variables, including market capitalization (*SIZE*), share turnover (*TURN*), and the indicator of firm-specific information (*D_TWTR*). The sample period is from December 2022 to December 2023, consisting of 271 days. Columns (1) to (4) use the full sample. Columns (5) to (8) use subsamples of stocks having non-zero *NET_BUY* or *NET_SELL* signals. Columns (1) to (4) (Columns (5) to (8)) use stocks having positive *NET_BUY* (*NET_SELL*). The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Sample	Full Sample				*NET_BUY*>0 or *NET_SELL*>0			
	Intraday Excess Return (bps)							
NET_SIG	1.198***	-0.276			0.972***	0.315		
	(3.41)	(-0.84)			(2.89)	(1.06)		
NET_BUY			1.608***	-0.562			0.847***	0.141
			(4.26)	(-1.20)			(3.02)	(0.43)
NET_SELL			-1.303	-2.720*			-1.303	-2.681*
			(-0.90)	(-1.73)			(-0.90)	(-1.83)
D_NET_SELL			6.356***	2.780			-2.591	2.790
			(3.00)	(1.31)			(-0.98)	(1.04)
SIZE		5.241***		5.403***		3.411***		3.607***
		(4.30)		(4.08)		(2.62)		(2.63)
TURN		-2.938***		-2.836***		-0.712		-0.482
		(-3.10)		(-3.10)		(-0.45)		(-0.31)
D_TWTR		-7.270		-5.612		-6.500		-5.738

Electronic copy available at: https://ssrn.com/abstract=4759311

		(-1.62)		(-1.36)		(-1.52)		(-1.37)
Constant	-9.498	-75.452***	-10.708	-77.309***	-3.776	-52.332**	-1.762	-54.665**
	(-1.44)	(-3.77)	(-1.63)	(-3.66)	(-0.57)	(-2.29)	(-0.28)	(-2.28)
#Days	271	271	271	271	271	271	271	271
Observations	1059956	1059956	1059956	1059956	283388	283388	283388	283388
R-sq	0.001	0.013	0.002	0.015	0.005	0.042	0.010	0.046

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Table 5. Fama-Macbeth Regression with Various Control Variables

This table reports the Fama-Macbeth regression results of regressing intraday excess returns on ChatGPT day trading signals with various control variables. The main dependent variable is intraday excess return, which is open-to-close price returns in basis point terms. The main independent variables are *NET_BUY* and *NET_SELL* signals. *NET_SIG* is the the difference between *SUM_BUY* and *SUM_SELL*. There are various control variables, including market capitalization (*SIZE*), share turnover (*TURN*), and an indicator of RIC information (*D_TWTR*). *MOM1M* is the previous 20 days of returns. *MOM12M* is the previous 12 to 1-month returns. *RET* $(t-1)$ is the previous day's returns. Panel B uses an indicator for Twitter mentioned and various market information as the main independent variables. *TURN* is the previous trading day's share turnover. *ILLIQ* is the Amihud (2002) illiquidity measure multiplied by 10,000. *BASPREAD* is the bid-ask spread, and *RETVOL* is the previous 20 days of return volatility. *INDRET* is the lagged industry portfolio's return, constructed by using SIC two-digit codes. Columns (1) and (2) use the full sample, and Columns (3) and (4) use stocks with non-zero *NET_BUY* or *NET_SELL* signals. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

	(1)	(2)	(3)	(4)
Sample	Full Sample		*NET_BUY*>0 or *NET_SELL*>0	
	Intraday Excess Returns (bps)			
NET_SIG	-0.203		0.200	
	(-0.74)		(0.79)	
NET_BUY		-0.359		0.025
		(-1.02)		(0.09)
NET_SELL		-2.061		-2.885**
		(-1.56)		(-2.28)
D_NET_SELL		2.815		3.425
		(1.37)		(1.41)
SIZE	3.328***	3.449***	2.220***	2.501***
	(4.09)	(3.92)	(2.61)	(2.76)
TURN	-0.981	-0.935	0.348	0.539
	(-1.18)	(-1.12)	(0.27)	(0.42)
D_TWTR	-5.622	-4.824	-5.191	-4.464
	(-1.40)	(-1.19)	(-1.26)	(-1.09)
MOM12M	0.049	0.046	0.013	0.005
	(1.03)	(1.00)	(0.25)	(0.10)
MOM1M	0.426***	0.424***	0.248*	0.238*
	(4.03)	(4.06)	(1.86)	(1.80)
RET $(t-1)$	1.246***	1.249***	0.571	0.593

Electronic copy available at: https://ssrn.com/abstract=4759311

	(3.85)	(3.88)	(1.01)	(1.05)
ILLIQ	7.553***	7.637***	5.955	5.947
	(3.57)	(3.56)	(1.12)	(1.12)
RETVOL	-0.129	-0.113	0.100	0.146
	(-0.22)	(-0.19)	(0.11)	(0.17)
BASPREAD	-1.924***	-1.924***	-1.910**	-1.923**
	(-5.20)	(-5.20)	(-2.46)	(-2.49)
INDRET	-0.301	-0.295	-1.184	-1.176
	(-0.23)	(-0.22)	(-0.75)	(-0.75)
Constant	-45.366***	-46.946***	-33.588**	-37.550**
	(-3.82)	(-3.71)	(-2.28)	(-2.39)
#Days	271	271	271	271
Observations	1034106	1034106	282891	282891
R-sq	0.035	0.036	0.092	0.095

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Table 6. Intraday Portfolio Performance of *Non-Ordered* ChatGPT Day Trading Signals

This table reports the intraday excess returns and CAPM alpha of the day trading strategy based on the non-ordered ChatGPT signal. We ask ChatGPT to provide lists of stocks to buy and sell using prompts without specifying to order stocks. A stock's *Non-Ordered BUY* (*SELL*) signal is constructed by *BUY* signals across the buy (sell) lists. The *Non-Ordered BUY* portfolio is constructed by weighted returns using the *Non-Ordered BUY* (*SELL*) signal. A stock's *Non-Ordered SUM_BUY* (*SUM_SELL*) signal is constructed by summing *Non-Ordered BUY* (*SELL*) signals. The *Non-Ordered NET_BUY* and *Non-Ordered NET_SELL* are constructed by netting the *Non-Ordered SUM_BUY* and the *Non-Ordered SUM_SELL* signals. Panel A reports results for *Non-Ordered BUY* and *SELL* signals. Panel B reports for *Non-Ordered SUM_BUY* and *SUM_SELL* signals. Panel C reports for *Non-Ordered NET_BUY* and *NET_SELL* signals. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. *Non-Ordered BUY* and *SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
BUY	*SELL*	*BMS*	*BUY*	*SELL*	*BMS*
-0.253	-0.836	0.584***	-1.582**	-2.160***	0.578***
(-0.18)	(-0.60)	(3.51)	(-2.16)	(-2.81)	(3.52)

Panel B. *Non-Ordered SUM_BUY* and *SUM_SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
SUM_BUY	*SUM_SELL*	*SUM_BMS*	*SUM_BUY*	*SUM_SELL*	*SUM_BMS*
1.193	0.010	1.182***	-0.160	-1.425*	1.265***
(0.93)	(0.01)	(3.02)	(-0.33)	(-1.93)	(3.33)

Panel C. *Non-Ordered NET_BUY* and *NET_SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
1.412	-1.886	3.298***	0.106	-3.384***	3.490***
(1.17)	(-1.09)	(3.49)	(0.25)	(-3.07)	(3.78)

Electronic copy available at: https://ssrn.com/abstract=4759311

Appendix Table 7. Intraday Portfolio Performance of *Order-Weighted* ChatGPT Day Trading Signals

This table reports the intraday excess returns and CAPM alpha of the day trading strategy based on the order-weighted ChatGPT signal. An order of a stock is in a unit interval, where 1 is the highest order. A stock has an order as 1 when it appears at the front of a buy or sell list generated by ChatGPT. A stock that appears at the end of a list has an order as 1/*n*. A stock's *Order-Weighted BUY* signal is constructed by averaging the orders in buy lists. The *Order-Weighted SELL* signal is similarly constructed. The *Order-Weighted BUY* portfolio is constructed by weighted returns using the *Order-Weighted BUY* signal, and the *Order-Weighted SELL* portfolio is similarly constructed. A stock's *Order-Weighted SUM_BUY* signal is constructed by summing all orders in the buy lists. The *Order-Weighted SUM_BUY* portfolio is constructed by weighting returns using the *Order-Weighted SUM_BUY* signal. The *Order-Weighted SUM_SELL* signal and portfolio are similarly constructed. The *Order-Weighted NET_BUY* is constructed by netting the *Order-Weighted SUM_BUY* and the *Order-Weighted SUM_SELL* signals. The *Order-Weighted NET_BUY* portfolio is constructed by weighting stock returns using the *Order-Weighted SUM_BUY* signals. The *Order-Weighted NET_SELL* portfolio is similarly constructed. Panel A reports results for *Order-Weighted BUY* and *SELL* signals. Panel B reports for *Order-Weighted SUM_BUY* and *SUM_SELL* signals. Panel C reports for *Order-Weighted NET_BUY* and *NET_SELL* signals. The standard errors are estimated using the Newey-West adjustment, and t-statistics are reported in parentheses. * ** , * *, * denote 1%, 5%, and 10% statistical significance.

Panel A. *Order-Weighted BUY* and *SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
BUY	*SELL*	*BMS*	*BUY*	*SELL*	*BMS*
-0.033	-0.530	0.498	-1.354**	-1.867**	0.514*
(-0.02)	(-0.36)	(1.62)	(-2.10)	(-2.22)	(1.68)

Panel B. *Order-Weighted SUM_BUY* and *SUM_SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
SUM_BUY	*SUM_SELL*	*SUM_BMS*	*SUM_BUY*	*SUM_SELL*	*SUM_BMS*
1.727	0.034	1.693**	0.346	-1.500	1.846***
(1.33)	(0.02)	(2.49)	(0.76)	(-1.64)	(2.82)

Panel C. *Order-Weighted NET_BUY* and *NET_SELL* Signal

Intraday Excess Return (monthly, %)			Intraday CAMP Alpha (monthly, %)		
NET_BUY	*NET_SELL*	*NET_BMS*	*NET_BUY*	*NET_SELL*	*NET_BMS*
1.640	-1.220	2.859**	0.310	-2.796**	3.106***
(1.32)	(-0.64)	(2.39)	(0.72)	(-2.29)	(2.68)

Electronic copy available at: https://ssrn.com/abstract=4759311