Chronologically Consistent Large Language Models*

Songrun He Linying Lv Asaf Manela Jimmy Wu

First draft: February 2025. This draft: March 2026.

Abstract

Large language models are increasingly used in social sciences, but their training data can introduce lookahead bias and training leakage. A good chronologically consistent language model requires efficient use of training data to maintain accuracy despite time-restricted data. Here, we overcome this challenge by training a suite of chronologically consistent large language models, ChronoBERT and ChronoGPT, which incorporate only the text data that would have been available at each point in time. Despite this strict temporal constraint, our models achieve strong performance on natural language processing benchmarks, outperforming or matching widely used models (e.g., BERT), and remain competitive with larger open-weight models. Lookahead bias is model and application-specific because even if a chronologically consistent language model has poorer language comprehension, a regression or prediction model applied on top of the language model can compensate. In an asset pricing application predicting next-day stock returns from financial news embeddings, we find that ChronoBERT and ChronoGPT's real-time outputs achieve Sharpe ratios comparable to a much larger Llama model, indicating that lookahead bias in this setting is modest. Our results demonstrate a scalable, practical framework to mitigate training leakage, ensuring more credible backtests and predictions across finance and other social science domains.

JEL Classification: G11, G12, G17

Keywords: Large language model, chronological consistency, lookahead bias, training leakage, backtesting

*We are grateful to Suproteem Sarkar (discussant), Will Mullins, Andy Neuhierl, Markus Pelger, and Mao Ye, as well as to conference participants at the 2025 NBER Summer Institute session on Big Data and High-Performance Computing for Financial Economics and seminar participants at the ABFR Webinar, Federal Reserve Board, University of Toronto, and Arizona State University for their insightful feedback and comments.

Songrun He is at Washington University in St. Louis (h.songrun@wustl.edu). Linying Lv is at Washington University in St. Louis (llyu@wustl.edu). Asaf Manela is at Washington University in St. Louis (amanela@wustl.edu). Jimmy Wu is at Washington University in St. Louis (jimmywu@wustl.edu).

> "Obviously, the time continuum has been disrupted, creating a new temporal event sequence resulting in this alternate reality."
>
> – Dr. Brown, *Back to the Future Part II*

1 Introduction

Large language models (LLMs) now permeate economics and finance, revealing nuanced patterns in unstructured text (Hoberg and Manela, 2025). They allow researchers to test hypotheses once considered unquantifiable. Yet these models often learn from data that did not exist at the historical moment. This "lookahead bias" (or training leakage) undermines empirical findings in settings that require real-time information (Glasserman and Lin, 2023; Sarkar and Vafa, 2024; Levy, 2024; Ludwig et al., 2025).

We address this challenge by training chronologically consistent LLMs exclusively on historical textual data available at the time. While training language models with historical data timestamped at the point of its availability is conceptually straightforward, ensuring these models are competitive with highly capable open-weight counterparts remains a significant challenge.

Building such models requires overcoming two primary obstacles: they require large amounts of computational power, and they rely on limited historical text. To tackle the computational side, we draw on efficient training methods (Warner et al., 2024; Jordan et al., 2024) to lower computing costs. To maximize information gained from a limited corpus, we follow Gunasekar et al. (2023) by selecting diverse, high-quality data, carefully filtered by publication date. This two-pronged strategy yields strong, chronologically consistent models even under tight resource constraints.

Our chief contribution is *ChronoBERT* and *ChronoGPT*: a pair of models for each year starting in 1999 that never see future data during training. On the GLUE benchmark

(Wang et al., 2019), both ChronoBERT and ChronoGPT outperform existing no-leakage models, including StoriesLM (Sarkar, 2024). Even our earliest ChronoBERT matches the popular BERT (Devlin et al., 2019), which ranks first in downloads among all language models on Hugging Face as of February 2025, and surpasses its finance-specific variant FinBERT (Huang et al., 2023).[1] Strong language understanding, we show, does not require data from the future. While ChronoBERT and ChronoGPT emerge as the strongest leak-free LLMs to date, they also maintain a modest scale, making them feasible for large embedding tasks where massive models (e.g., Llama) are too costly.

We validate the chronological consistency of our models through a systematic test of information leakage. Beyond targeted exercises that probe knowledge of specific events and presidents, we compute the probability that each ChronoGPT vintage assigns to 395 neologisms introduced between 2002 and 2022. The average probability is centered precisely at zero before each word's first known use date and jumps significantly only afterward, providing systematic evidence consistent with minimal leakage in our training corpora.

Among all social sciences, finance is particularly sensitive to lookahead bias. Market efficiency tests (Fama, 1970) assume that prices reflect only facts known at the time. A model that "time-travels" by reading tomorrow's news distorts such tests. While researchers could keep a held-out sample of recent data to avoid lookahead bias, the limited panel of asset prices commonly studied forces most studies to rely on backtesting. Hence, a chronologically inconsistent language model can bias measures of risk and market inefficiency.

We test for lookahead bias in a news-based return forecasting strategy with ChronoBERT and ChronoGPT. Leveraging extensive financial newswire data, we find that the portfolio performance of ChronoBERT and ChronoGPT matches that of Llama 3.1 (Dubey et al.,

[1]Based on download statistics from Hugging Face at https://huggingface.co/models?sort=downloads.

2024), with both models delivering economically substantial and statistically significant gains compared to StoriesLM (Sarkar, 2024) and FinBERT (Huang et al., 2023). Our findings suggest that lookahead bias in this specific setting is relatively modest.

To establish that this conclusion holds beyond our baseline setup, we conduct robustness checks along two key dimensions: model scale and forecasting horizon. First, we examine the effect of model size by scaling our architectures across a 12-fold range, directly addressing the concern that larger parameter counts might aggressively amplify lookahead bias. Second, we extend our return prediction horizons from a single day to periods spanning five days up to one quarter. Across all of these experiments, we find no significant performance gaps between our chronologically consistent real-time models and their inconsistent counterparts that possess future knowledge. These findings reinforce our main result, suggesting that lookahead bias remains modest in the context of embedding-based return prediction.

An important observation is that the impact of lookahead bias is model- and application-specific. In this setup, while ChronoBERT and ChronoGPT may exhibit lower language comprehension on general tasks compared to unconstrained models, downstream predictive models built on top of our models can adapt to these limitations, mitigating potential drawbacks in financial forecasting. However, lookahead bias could manifest more severely in other use cases, such as generative tasks or direct prompting. Therefore, our conclusion that lookahead bias is modest applies specifically to return forecasting using news embeddings, and researchers should exercise caution when generalizing to other applications.

Finally, we uncover a novel "envelope" phenomenon in trading performance: real-time models tend to outperform both earlier and later vintage models when applied to their own time period. To understand this pattern, we compute perplexity scores for each ChronoGPT vintage on each year of the Dow Jones Newswire corpus, evaluating three

candidate channels: language understanding, catastrophic forgetting, and contextual knowledge. For every corpus year, we find that most later vintages achieve lower perplexity than the real-time model, ruling out the language channel. Perplexity also decreases with vintage year, ruling out catastrophic forgetting. Yet it is the real-time model that generates the highest trading returns. This points to the contextual knowledge channel as the most plausible driver: later models exhibit better language understanding but lack sensitivity to the contemporaneous context needed to value financial signals as effectively.

Our contribution is threefold. First, we explain how to build chronologically consistent LLMs that preserve validity in economic and financial tests. Second, we find that strong language understanding can be attained without introducing training leakage: ChronoBERT and ChronoGPT remain competitive with powerful, open-weight baselines. Third, we measure lookahead bias in a news-based return forecasting setting with embeddings and find it modest. These points open a path to more credible LLM applications across the social sciences.

Our work is related to two broad strands of literature. First, a large literature studies how news forecasts stock returns (e.g., Tetlock et al., 2008; Jiang et al., 2021; Ke et al., 2019). Early dictionary-based or word-count methods measured only coarse signals, delivering smaller economic effects than modern LLMs. LLMs have pushed text-based stock return forecasting further. Chen et al. (2023) show that news embeddings from LLMs strongly predict next-day returns. Lopez-Lira and Tang (2023) demonstrate how prompting LLMs yields robust signals. These results highlight the growing potency of advanced NLP in finance.

Our main contribution to this literature is to show that the robust return predictability achieved through LLMs is not driven by lookahead bias, thereby addressing a critical concern in the use of these advanced models for financial forecasting.

A second strand of literature pertains to the application and development of natural language processing (NLP) tools for financial economics research (Hoberg and Manela, 2025). Methodological advancements in this area have evolved from early dictionary-based approaches (Tetlock, 2007; Loughran and McDonald, 2011), to text regressions (Manela and Moreira, 2017; Kelly et al., 2021), to topic modeling (Bybee et al., 2024), to neural embedding methods (Goetzmann et al., 2022), and most recently, to the integration of LLMs (Jha et al., 2025; Chen et al., 2023; Lv, 2024). These developments underscore the growing demand for more advanced and scalable tools to answer research questions in finance and economics.

In this broader context, our main contribution is a framework for developing LLMs that are both free of lookahead bias and capable of high-level language comprehension. Our approach does not require masking (Glasserman and Lin, 2023; Engelberg et al., 2025), which can destroy information, and may not completely remove the lookahead bias of high-memory LLMs (Levy, 2024; Lopez-Lira et al., 2025; Didisheim et al., 2025). Instead, we train a series of chronologically consistent models with knowledge cutoffs from 1999 to 2024 that offer superior language understanding and more up-to-date knowledge than similar attempts (e.g., Sarkar, 2024).[2]

The rest of the paper is organized as follows: Section 2 describes our approach and data for model pretraining and evaluation. Section 3 presents the empirical performance of our model. Section 4 concludes.

[2]In contemporaneous work, Rahimikia and Drinkall (2024) introduce a series of BERT LLMs trained on time-stamped proprietary financial texts for predictions. They do not evaluate the models' general language understanding capabilities, but based on the financial forecasting metrics they do provide, their best-performing model generates a long-short portfolio Sharpe ratio of 3.45 from 2017 to 2023, significantly underperforming ChronoGPT and ChronoBERT. Unfortunately, the authors have since retracted their models from Hugging Face, preventing further direct comparison. Separately, Drinkall et al. (2024) introduce a series of GPT LLMs trained on a time-stamped corpus comprising news and Wikipedia articles. Their models are smaller in size and achieve a substantially lower HellaSwag score of 0.25, which is indistinguishable from random guessing.

2 Methodology and Data

In this section, we outline the methodology we use for model pretraining and describe the approach for evaluating their performance. Specifically, we assess their ability in both language understanding and asset pricing tasks.

2.1 Pretraining Methodology for ChronoBERT

When pretraining ChronoBERT, we incorporate a modern BERT architecture from Portes et al. (2023) and Warner et al. (2024).[3] Compared to the original BERT model by Devlin et al. (2019), this enhanced architecture integrates recent advancements in rotary positional embeddings that support longer context lengths and employ flash attention, significantly improving pretraining efficiency and computational speed.

For the pretraining task, we follow Warner et al. (2024) by adopting masked token prediction while omitting the next sequence prediction task, as prior research has shown the latter increases training overhead without meaningful performance gains.[4]

The quality of pretraining data is critical to achieving BERT-level performance. Gunasekar et al. (2023) demonstrate that using high-quality, "textbook-like" data leads to faster convergence and improved model outcomes. Motivated by this insight, we filter our pretraining corpus using the FineWeb-edu classifier from Penedo et al. (2024), retaining only texts with scores above two.[5]

However, restricting the corpus to texts with historical dates—particularly from early historical periods—introduces data scarcity challenges. Muennighoff et al. (2023) explore

[3]We thank the authors for providing their pretraining code at https://github.com/mosaicml/examples/tree/main/examples/benchmarks/bert and https://github.com/AnswerDotAI/ModernBERT.

[4]We provide details on ChronoBERT's masked language modeling pretraining objective and attention mechanisms in Appendix C and contrast them with those of ChronoGPT.

[5]We thank the authors for providing the classifier at https://huggingface.co/HuggingFaceFW/fineweb-edu-classifier.

the scaling laws of LLMs under data constraints, highlighting the benefits of iterative training on limited high-quality data. Following their insights, we train our model over multiple epochs to maximize learning from the available corpus. Our first model checkpoint ChronoBERT$_{1999}$ is trained on 460 billion tokens, with more than 70 epochs through the dataset, when the validation loss stops improving.

2.2 Pretraining Methodology for ChronoGPT

ChronoGPT is pretrained using a modified nanoGPT architecture, incorporating key enhancements from Jordan et al. (2024).[6] Compared to the original GPT-2 implementation by Radford et al. (2019), this architecture integrates several optimizations, including Rotary embeddings, QK-Norm, skip connection, and FlexAttention, improving both computational efficiency and scalability.

ChronoGPT is trained using a causal language modeling (CLM) objective, learning to predict the next token based on previously observed sequences.[7]

To determine the optimal model size, we first conducted a scaling law experiment to assess the impact of parameter count within our data-constrained, chronological training regime. As detailed in Appendix B, this test revealed that increasing model size from 124M to 1.5B consistently improved performance on both validation loss and language understanding. Based on these findings, we selected the 1.5B parameter model, as it offered the best performance while remaining within our computational budget.

Given the efficiency gains provided by the modded-nanoGPT optimizer, ChronoGPT requires fewer training iterations to reach competitive performance. We adopt a low-epoch, high-throughput strategy, allowing the model to rapidly extract essential linguistic

[6]We thank the authors for providing their training framework at https://github.com/KellerJordan/modded-nanogpt.

[7]We provide details on ChronoGPT's causal language modeling pretraining objective and attention mechanisms in Appendix C and contrast them with those of ChronoBERT.

and contextual structures. Our first vintage model, ChronoGPT$_{1999}$, is trained on 71 billion tokens over approximately 10 epochs, when validation loss stops improving, striking a balance between computational efficiency and representational robustness.

2.3 Evaluation Methodology

To evaluate the performance of our models, we assess both language understanding capabilities and economic forecasting performance. We also apply the same evaluation methodology to several other LLMs for benchmarking.

2.3.1 Language Understanding

To assess our models' language understanding abilities, we employ distinct evaluations that align with the fundamental architectures of ChronoBERT and ChronoGPT. ChronoBERT is built on an encoder-based structure that creates a bidirectional representation of all input tokens, making it an ideal candidate for classification benchmarks such as GLUE. In contrast, ChronoGPT leverages a generative, autoregressive framework that is naturally suited to open-ended and commonsense reasoning tasks. Accordingly, we compare ChronoBERT against BERT on GLUE and compare ChronoGPT against GPT-2 on HellaSwag, ensuring that each model is assessed on the tasks best suited to its core strengths.

The GLUE evaluation framework, introduced by Wang et al. (2019), comprises multiple classification tasks designed to measure a model's language understanding.[8] This framework was also the primary evaluation metric used in Devlin et al. (2019) to assess BERT's language capabilities.

[8]The details and leaderboard of GLUE evaluation can be found at https://gluebenchmark.com/. Appendix A provides an overview of the eight GLUE tasks. Appendix C describes the architectural differences between ChronoBERT and ChronoGPT for sequence classification in GLUE evaluation.

Following pretraining, we further fine-tune the model on task-specific training datasets and evaluate its performance on a held-out test set.

For fine-tuning, we adopt the training specifications and hyperparameters outlined in Warner et al. (2024). Among the eight GLUE tasks, RTE, MRPC, and STS-B are initialized from the MNLI checkpoint to enhance performance.

Beyond GLUE, we evaluate ChronoGPT's commonsense reasoning with HellaSwag (Zellers et al., 2019) in a zero-shot setting. HellaSwag presents a sentence completion task where each context ends with an incomplete phrase (e.g., "A woman sits at a piano,") and the model must choose the most plausible continuation from four possible endings (e.g., "She sets her fingers on the keys."). We compute the likelihood of each candidate ending under ChronoGPT's autoregressive predictions and select the most likely continuation. Since HellaSwag employs adversarial filtering to create highly deceptive distractors, strong performance on this task signals robust contextual and procedural reasoning.[9]

2.3.2 Predicting Stock Returns using Financial News

We investigate whether improved language understanding translates into economic gains by using different language models to predict stock returns from financial news. Based on these predictions, we construct portfolios and evaluate the performance of long-short strategies.

Following Chen et al. (2023), we first aggregate all news articles' headlines for a stock on a given trading day together. Next, we transform this text into embeddings. Specifically, we process each piece of text through different language models and extract the hidden

[9]The leaderboard and data of HellaSwag can be found at https://paperswithcode.com/sota/sentence-completion-on-hellaswag. Appendix A provides detailed information on the methodology used for HellaSwag evaluations.

states of all tokens.[10] The final embedding for each text is obtained by averaging the token embeddings.[11]

Next, we link each news article to stock returns on the following trading day and fit a Fama-MacBeth regression with a ridge penalty to map news embeddings to return predictions. Each month m, we estimate the following cross-sectional ridge regression:

$$r_{i,t+1} = \alpha_m + \beta_m' e_{i,t} + \varepsilon_{i,t+1}, \quad \text{for } i = 1, \cdots N \text{ and } t = 1 \cdots T,^{12} \tag{1}$$

where $e_{i,t}$ represents the embedding of all news for firm i on day t. To construct real-time out-of-sample forecasts, in month m', we use an average of forecasts over all previous months' cross-sectional models:

$$\hat{r}_{i,t+1} = \bar{\alpha}_{m'} + \bar{\beta}_{m'}' e_{i,t}, \quad \text{for } i = 1, \cdots N \text{ and } t = 1 \cdots T, \tag{2}$$

where $\bar{\alpha}_{m'} = \frac{1}{m'-1} \sum_{m=1}^{m'-1} \hat{\alpha}_m$ and $\bar{\beta}_{m'} = \frac{1}{m'-1} \sum_{m=1}^{m'-1} \hat{\beta}_m$.

Using these out-of-sample predictions, we sort stocks into decile portfolios at the end of each trading day, based on forecasts from different language models. We then evaluate the performance of daily-rebalanced long-short decile portfolios constructed from these predictions.

[10] Different models generate the hidden states in distinct ways, which can impact predictive performance. To account for this, we extract three versions of token embeddings: (1) using the hidden states from the last layer, (2) averaging the hidden states across all layers, and (3) using the hidden states from the first layer. We determine in real time which approach yields the highest 'H-L' portfolio Sharpe Ratio over an expanding historical window and use that for forecasting. We report, by model, the fraction of months in which each embedding strategy is selected (Table D.3) in the Appendix.

[11] Coleman (2020) provides a rationale for averaging token embeddings to get sequence embeddings.

[12] We use a leave-one-out cross-validation algorithm to determine the ridge penalty λ chosen from grid points ranging from 10^{-10} to 10^{10}.

2.4 Data

In this section, we present the data we used for pretraining our language models and the financial newswire data we used to evaluate our language models.

2.4.1 Pretraining Data

Our pretraining strategy is split into two stages: (1) the pre-1999 initial pretraining stage and (2) the post-1999 incremental pretraining stage.

Table 1 documents the datasets we used for pretraining our models. ChronoBERT$_{1999}$ and ChronoGPT$_{1999}$ are pretrained for multiple epochs on a corpus consisting of peS2o[13], American Stories[14], and Newswire[15]. These datasets comprise chronologically organized English text sourced from diverse domains, including archived news articles and scientific publications. For each text in these databases, there are detailed publication dates, allowing us to include only texts published before the year 2000, ensuring a focus on no leakage of future knowledge.

After further quality filtering, the final pre-1999 dataset consists of 7 billion tokens of high-quality text. The leftmost bar in Figure 1 shows the composition of the pre-1999 corpus. To maximize learning from the available corpus, we iterate training on the corpus as many times as possible until we observe a flattening in language understanding from both ChronoBERT$_{1999}$ and ChronoGPT$_{1999}$.

For incremental pretraining, we rely on peS2o, Wikipedia, and FineWeb[16] datasets, whose time span overlaps with later historical periods, to update the knowledge of the models. ChronoBERT's incremental pretraining uses peS2o and FineWeb, while

[13]Dataset downloaded from https://huggingface.co/datasets/allenai/peS2o.

[14]Dataset downloaded from https://huggingface.co/datasets/dell-research-harvard/AmericanStories.

[15]Dataset downloaded from https://huggingface.co/datasets/dell-research-harvard/newswire.

[16]Dataset downloaded from https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu. We use the sample-10BT subset of the dataset.

Dataset	Description	Literature	Time Span	Timestamp Source
peS2o	Open-access academic papers	Lo et al. (2020)	1970–2022	Publication
American Stories	Full-text articles from historical U.S. newspapers	Dell et al. (2023)	1774–1963	Publication
Newswire	U.S. newswire articles from local newspapers	Silcock et al. (2024)	1878–1977	Publication
Wikipedia	Wikipedia articles	—	2001–2024	Revision
FineWeb	Filtered Common Crawl data	Penedo et al. (2024)	2013–2024	Crawl

Table 1 Pretraining Datasets

This table summarizes the datasets used for pretraining, including their descriptions, associated literature, time spans, and how timestamps were derived.

ChronoGPT additionally incorporates Wikipedia. For the Wikipedia dataset, we download and compile a history of revisions for all Wikipedia articles to ensure strict no leakage based on the revision dates. For the FineWeb dataset, which is derived from Common Crawl, we use the crawl date to avoid lookahead bias. After quality filtering, the final incremental pretraining dataset consists of 112 billion tokens of high-quality, diverse text. Figure 1 presents the token compositions for each year in our incremental pretraining sample.

2.4.2 Financial Newswire Data

We utilize the Dow Jones Newswire dataset, a real-time newswire subscribed to by major institutional investors. This dataset provides extensive coverage of financial markets, economies, and companies, aggregating reports from leading media sources such as the Wall Street Journal, Barron's, and MarketWatch.

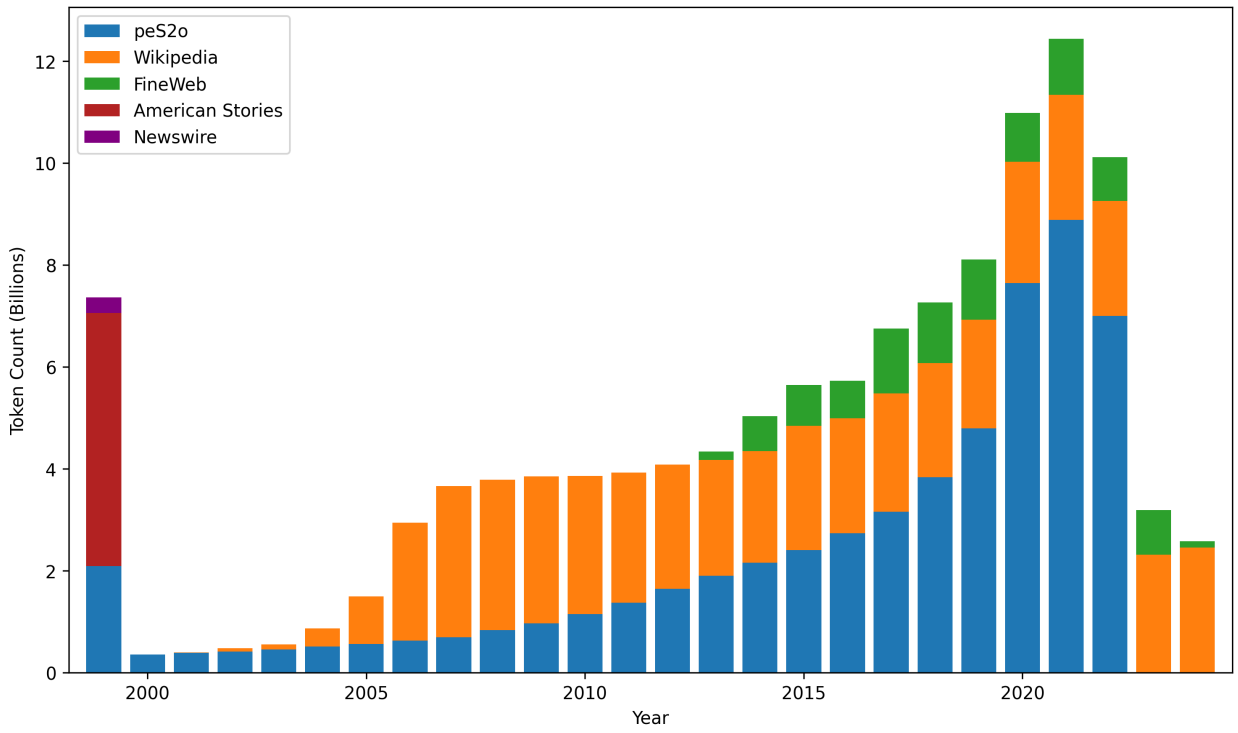


Figure 1 Incremental Token Count Across Model Years

This figure displays the yearly stacked token counts for the five pretraining datasets: peS2o, Wikipedia, FineWeb, American Stories, and Newswire. A detailed description of each dataset is provided in Table 1.

The dataset includes news headlines, full article texts, and precise display timestamps, with microsecond-level accuracy for when the news becomes available to clients. Following Ke et al. (2019), we focus on firm-specific news that can be attributed to a single company. For each firm-day observation, we aggregate all news headlines related to the firm within the trading day window—spanning from 4:00 p.m. EST on day $t-1$ to 4:00 p.m. EST on day t—and treat the combined text as the firm's textual information. Each concatenated set of headlines is then processed through our embedding framework to generate numerical text representations.

After the embedding step, we merge the news dataset with close-to-close returns on trading day $t+1$ from CRSP together to examine the predictability of stock returns using LLMs trained with real-time available textual data.

Our dataset covers the period from January 2007 to July 2023. The first year serves as a burn-in period to estimate the initial return prediction model, resulting in a final asset pricing test sample spanning January 2008 to July 2023.

3 Results

In this section, we first describe the pretraining process, focusing on validation loss and language understanding as the model is trained on an increasing number of tokens. We then benchmark the language understanding capabilities of our ChronoBERT and ChronoGPT against several other language models. We then validate the chronological consistency of our models. Finally, we assess their asset-pricing performance in the news return prediction exercise.

3.1 Pretraining Model Fit

Firstly, on pretraining performance, our results confirm the scaling law proposed by Muennighoff et al. (2023) in a data-limited environment. As shown in Figure 2, validation loss (measured via cross-entropy) decreases consistently as the number of training tokens increases.[17] Simultaneously, language prediction accuracy improves with training progress.

These improvements translate into enhanced language understanding, as reflected in the GLUE and HellaSwag scores. ChronoBERT surpasses the original BERT's performance on the GLUE benchmark after being trained on approximately 350 billion tokens. Similarly, ChronoGPT outperforms GPT-2 on the HellaSwag benchmark after 3.15 billion training

[17]For ChronoBERT, we use a subset of the C4 corpus from https://huggingface.co/datasets/allenai/c4 as the validation set. The C4 data is a cleaned version of the Common Crawl data. For ChronoGPT, we adopt the FineWeb validation set, following the setup of https://github.com/KellerJordan/modded-nanogpt.

tokens.

When evaluated against more demanding benchmarks, the performance of our models begins to plateau. We compare ChronoBERT to ModernBERT, which is trained on a significantly larger text corpus, and ChronoGPT to GPT-2 XL, which has a comparable number of parameters. ChronoBERT's GLUE score stabilizes at 85, falling short of ModernBERT's 88. Likewise, ChronoGPT achieves a HellaSwag score of 37% after 40 billion tokens, which is lower than the 48% achieved by GPT-2 XL. We attribute this performance ceiling to the temporally-constrained nature of our training data, as continued training on the same dataset yields no further improvements.

Starting from these high-quality base models in the early period, we continue incremental pretraining using textual data afterward. We create model checkpoints for each year from 1999 to 2024 (26 models in total). Figure 3 presents our models' validation loss and language understanding scores as we train with incremental textual data over time.

We find that with the introduction of new data, the validation loss continues to drop. Starting in the year 2013, we introduce high-quality common crawl data. We witness a significant decrease in validation loss with the increase in data diversity. In the right panel of Figure 3, the GLUE and HellaSwag scores for nearly all models exceed those of BERT and GPT-2, highlighting their superior language understanding and overall quality.[18]

3.2 Language Understanding

We further compare language understanding against other models. Table 2 summarizes key characteristics of these models, including parameter counts, context lengths, and knowledge cutoffs.

The models evaluated include:

[18]ModernBERT is trained on roughly 1.8 trillion tokens, over three times the 525 billion tokens used for ChronoBERT$_{2024}$. This much larger pretraining corpus largely explains its stronger GLUE performance.

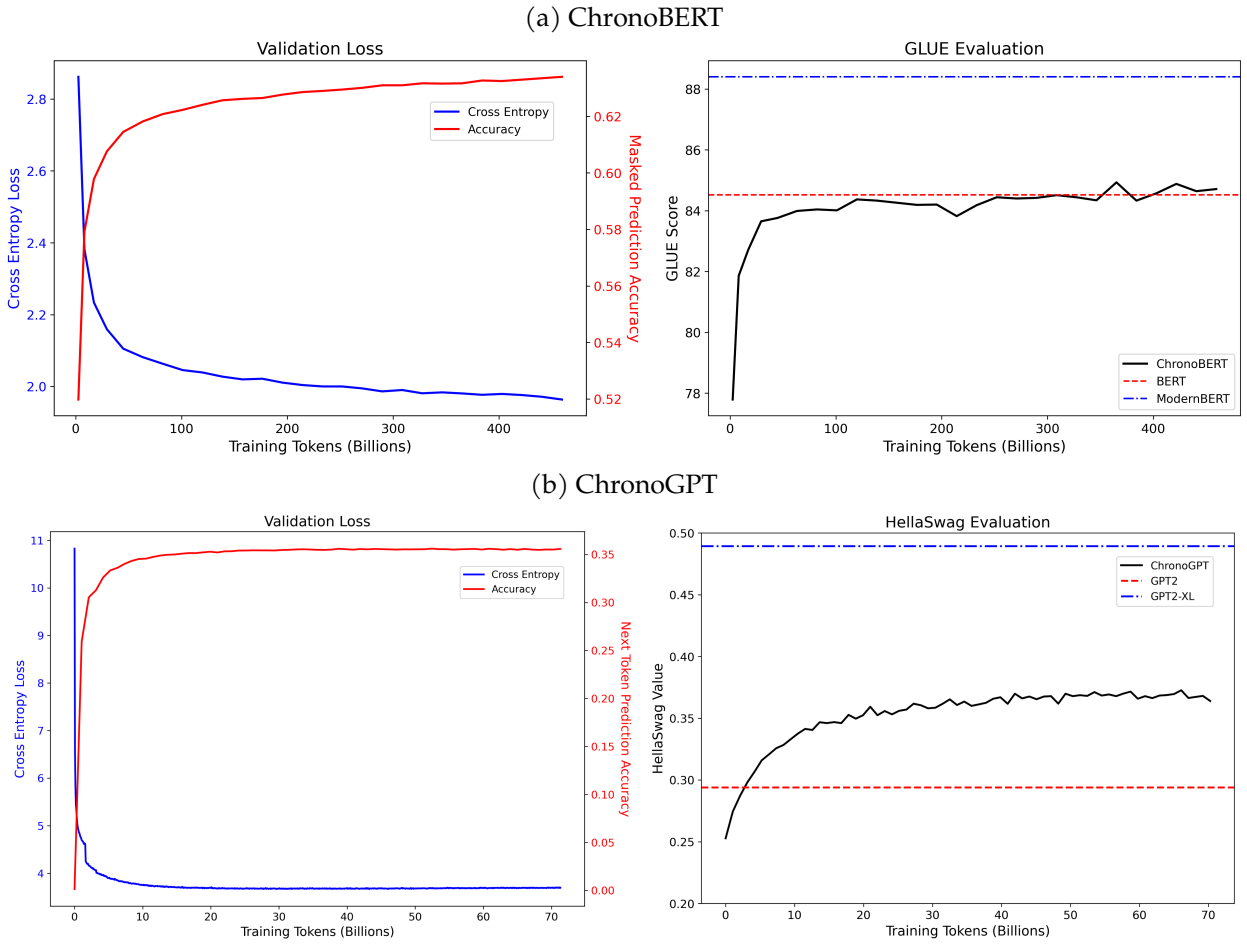
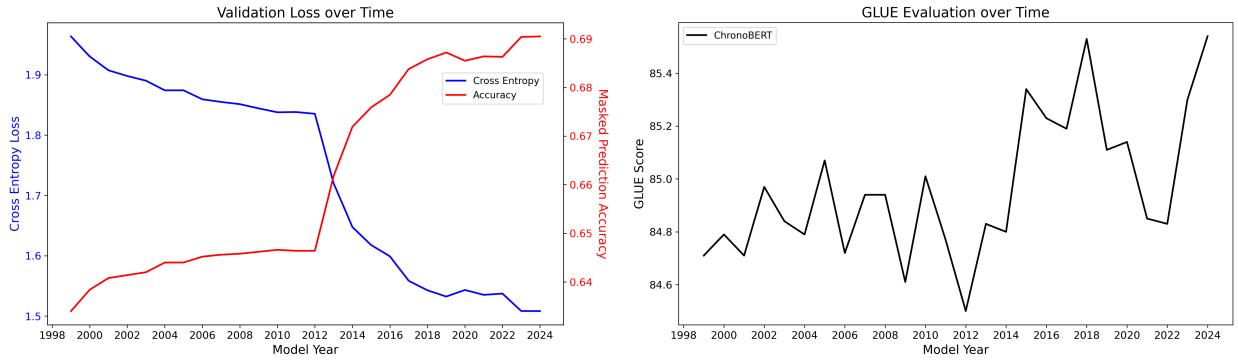


Figure 2 Validation Loss and Evaluation Scores versus Pretraining Tokens

The left panel shows the validation loss, measured using cross-entropy loss and language prediction accuracy, as the ChronoBERT$_{1999}$ and ChronoGPT$_{1999}$ are trained on an increasing number of tokens. The right panel displays the GLUE scores and HellaSwag values as training progresses. For ChronoBERT and ChronoGPT, the final model checkpoint is trained on 460 billion tokens and 70 billion tokens, respectively. The training corpus consists of text up to December 1999.

ChronoBERT$_t$: Our initial BERT-based model, ChronoBERT$_{1999}$ is pretrained on 460 billion tokens of pre-2000, diverse, high-quality, and publicly available text data listed in Table 1 and Figure 1 (peS2o, American Stories, and Newswire) to ensure no leakage of data afterward. Then, each year t starting in 2000, we start from the model trained in the previous year and continue training it on the data available in year t. The incremental pretraining datasets include peS2o and FineWeb. Our final checkpoint of the ChronoBERT series, ChronoBERT$_{2024}$, is pretrained on 525 billion tokens of diverse, high-quality, and

(a) ChronoBERT



(b) ChronoGPT

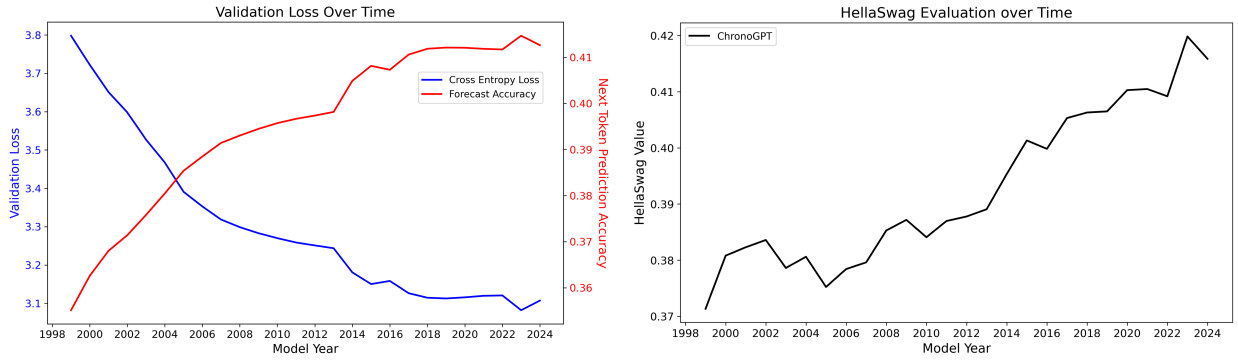


Figure 3 Validation Loss and Evaluation Scores over Time

The left panel shows the validation loss, measured using cross-entropy loss and masked language prediction accuracy as the model is trained over time. The right panel displays the GLUE and HellaSwag scores as training progresses over time.

publicly available text until December 2024.

ChronoGPT$_t$: Our initial GPT-based model, ChronoGPT$_{1999}$, is pretrained with the same chronologically consistent text data as ChronoBERT, comprising 70 billion tokens. By restricting the training data to information available up to each respective point in time, the model ensures no lookahead bias. The ChronoGPT series spans from 1999 to 2024, with each yearly checkpoint building upon the previous year's model. The incremental pretraining datasets include peS2o, Wikipedia, and FineWeb.

Llama 3.1: The 8B-parameter variant of the Llama 3.1 model by Dubey et al. (2024). [19]

[19]Model downloaded from https://huggingface.co/meta-llama/Llama-3.1-8B.

BERT: The original BERT model trained on Wikipedia and the BookCorpus dataset by Devlin et al. (2019).[20]

FinBERT: A domain-specific model pretrained on financial texts, including regulatory filings, analyst reports, and earnings call transcripts, from Huang et al. (2023).[21]

StoriesLM: A pretrained model from Sarkar (2024), trained on historical news articles. We use the final version trained on data up to 1963.[22]

	Parameters	Context Tokens	Knowledge Cutoff
ChronoBERT_{1999}	149M	1,024	December, 1999
⋮	⋮	⋮	⋮
ChronoBERT_{2024}	149M	1,024	December, 2024
ChronoGPT_{1999}	1,552M	1,792	December, 1999
⋮	⋮	⋮	⋮
ChronoGPT_{2024}	1,552M	1,792	December, 2024
Llama 3.1	8,030M	128,000	December 2023
GPT2-XL	1,542M	1024	February 2019
ModernBERT	149M	8192	December 2024
BERT	110M	512	October, 2018
FinBERT	110M	512	December, 2019
StoriesLM	110M	512	December, 1963

Table 2 Characteristics and Knowledge Cutoffs of Different LLMs

This table provides an overview of our chronologically consistent large language models (ChronoBERT and ChronoGPT) as well as several natural benchmark models, including their number of parameters, maximum context length, and knowledge cutoff dates.

Table 3 displays the GLUE scores for ten different models. While Llama 3.1 achieves the highest overall GLUE score (86.31), ChronoBERT_{2024} and ChronoBERT_{1999}, standing at less than 1/50 of Llama's size, still deliver strong performance, surpassing BERT (84.52) while substantially outperforming StoriesLM (79.36) and FinBERT (76.41). Notably, both ChronoBERT models Pareto dominate the last two models—they outperform StoriesLM

[20]Model downloaded from https://huggingface.co/google-bert/bert-base-uncased.
[21]Model downloaded from https://huggingface.co/yiyanghkust/finbert-pretrain.
[22]Model downloaded from https://huggingface.co/StoriesLM/StoriesLM-v1-1963.

	ChronoBERT_{1999}	ChronoBERT_{2024}	ChronoGPT_{1999}	ChronoGPT_{2024}	Llama 3.1
COLA	57.32	56.32	35.50	48.20	55.86
SST2	91.82	92.58	90.60	92.40	95.49
MRPC	92.71	92.45	87.66	88.55	88.22
STSB	89.57	89.93	80.65	86.07	90.67
QQP	88.54	88.90	83.70	85.75	89.67
MNLI	86.19	86.89	82.27	84.91	89.59
QNLI	90.61	92.04	84.68	88.56	95.35
RTE	80.94	85.20	74.30	78.63	85.63
GLUE	84.71	85.54	77.42	81.63	86.31
	GPT2-XL	ModernBERT	BERT	FinBERT	StoriesLM
COLA	51.48	65.10	57.59	28.99	46.85
SST2	93.00	96.00	92.62	89.03	90.44
MRPC	88.39	92.20	90.76	88.59	89.33
STSB	90.24	91.80	90.07	85.72	87.01
QQP	88.05	92.10	88.21	86.60	86.88
MNLI	86.96	89.10	84.98	79.23	79.78
QNLI	92.46	93.90	91.52	86.12	87.44
RTE	83.03	87.40	80.43	67.00	67.15
GLUE	84.20	88.40	84.52	76.41	79.36

Table 3 GLUE Score Evaluations for Different LLMs

This table compares the GLUE benchmark scores of our chronologically consistent large language models (ChronoBERT and ChronoGPT) as well as several natural benchmark models. Tasks are grouped into three categories: (1) Single-sentence classification (COLA, SST2), (2) Paraphrase/semantic similarity (MRPC, STSB, QQP), and (3) Natural language inference (MNLI, QNLI, RTE). The final row shows the average GLUE score across all tasks.

(designed to avoid lookahead bias) and FinBERT (domain-specific) across all individual tasks, with particularly large margins on COLA, RTE, and MNLI. This performance advantage persists even in BERT-competitive tasks like MRPC and QQP.

Similarly, the ChronoGPT models showcase the effectiveness of our training methodology, with ChronoGPT_{2024} (81.63) significantly outperforming FinBERT and StoriesLM. The most salient finding for ChronoGPT is its growth; it gains over 4 points on the GLUE benchmark from its 1999 counterpart (77.42), with especially large improvements in grammatical correctness (COLA) and inference (RTE). This confirms that chronological

data enrichment progressively enhances language understanding in both encoder-only and decoder-only architectures.

Overall, ChronoBERT and ChronoGPT exhibit strong language understanding, with ChronoBERT coming remarkably close to Llama 3.1's performance despite Llama 3.1 being a much larger and more recent model. The performance gap between ChronoBERT and StoriesLM likely stems from differences in training scale (460B versus 19B tokens) and the quality and diversity of the training data (ChronoBERT's high-quality corpus versus StoriesLM's unfiltered news-only dataset). Similarly, ChronoBERT and ChronoGPT's significant edge over FinBERT highlights the importance of diverse and high-quality pretraining data, as FinBERT's domain-specific financial texts lack comprehensive quality checks.

Importantly, our chronologically consistent models demonstrate that eliminating lookahead bias need not come at a cost of the core capabilities expected of modern LLMs. This makes them particularly valuable for applications that require good language understanding and no lookahead bias. By maintaining both strong language understanding and temporal integrity, our models are particularly well-suited for applications where preserving the timeline of information is critical, making them a robust choice for a wide range of downstream tasks.

3.3 Validation of Chronological Consistency

Although we exercise great caution in curating our training data to ensure they include only information that we believe was available by a specific date, this process is not necessarily without imperfections. Potential inaccuracies in recorded publication dates—for example, if a hard copy document was digitized via optical character recognition (OCR) and an incorrect date was captured in the process—may result in leakage: the unintended inclusion of information that was not actually available at the intended time.

To detect leakage, we employ two complementary strategies. First, we perform targeted evaluations probing whether each model vintage can predict discrete, date-specific facts such as U.S. presidential elections and major world events beyond its knowledge cutoff. Second, we conduct a broader test based on 395 neologisms with known introduction dates, examining whether the probability each ChronoGPT vintage assigns to these terms is near zero before their first known use and rises sharply only afterward.

3.3.1 Token Prediction of Future Events

We construct a textual sequence with a missing token representing time-specific information beyond the model's cutoff. We use each model to tokenize the sentence and predict the missing token. Due to the bi-directional nature of ChronoBERT, the missing token is a masked token in the middle of a sentence, with the surrounding language giving context for the prediction. Consider the following sentence involving an incoming U.S. president, where the president's name is masked and is the target of the prediction:

> "After the {year} U.S. presidential election, President [MASK] was inaugurated as U.S. President in the year {year+1}."

As for ChronoGPT, due to its autoregressive nature, the missing token is at the end of a textual sequence, with the preceding context guiding the prediction. We construct a textual sequence that is essentially a few-shot learning exercise. Denoting the most recent president as the p^{th} president, consider the following list involving the incoming U.S. president and his three most recent predecessors, where the name of the most recent president (name_p) is omitted and is the target of the prediction:

> "U.S. Presidents in chronological order:
> Took office in $\{\text{year}_{p-3}+1\}$: President $\{\text{name}_{p-3}\}$
> Took office in $\{\text{year}_{p-2}+1\}$: President $\{\text{name}_{p-2}\}$

	Election year						Accuracy	
	1992	2000	2008	2016	2020	2024	Pre-cutoff	Post-cutoff
Correct output	Clinton	Bush	Obama	Trump	Biden	Trump		
BERT	**Clinton**	Clinton	**Obama**	Obama	Obama	Obama	2/4	0/2
ModernBERT	**Clinton**	**Bush**	**Obama**	**Trump**	**Biden**	**Trump**	5/5	1/1
$\text{ChronoBERT}_{\text{Realtime}}$	—	Hoover	Bush	Obama	Trump	Biden	—	0/5
ChronoBERT_{1999}	Roosevelt	Hoover	Roosevelt	Hoover	Hoover	Washington	0/1	0/5
ChronoBERT_{2000}	**Clinton**	Clinton	Clinton	Clinton	Clinton	Wilson	1/2	0/4
ChronoBERT_{2001}	**Clinton**	Clinton	Clinton	Clinton	Clinton	Washington	1/2	0/4
ChronoBERT_{2002}	**Clinton**	**Bush**	Bush	Clinton	Bush	Clinton	2/2	0/4
ChronoBERT_{2003}	**Clinton**	**Bush**	Bush	Clinton	Bush	Clinton	2/2	0/4
ChronoBERT_{2004}	**Clinton**	**Bush**	Bush	Clinton	Bush	Clinton	2/2	0/4
ChronoBERT_{2005}	**Clinton**	**Bush**	Bush	Bush	Bush	Clinton	2/2	0/4
ChronoBERT_{2006}	**Clinton**	**Bush**	Bush	Bush	Bush	Clinton	2/2	0/4
ChronoBERT_{2007}	**Clinton**	**Bush**	Bush	Bush	Bush	Monroe	2/2	0/4
ChronoBERT_{2008}	**Clinton**	**Bush**	Bush	Obama	Bush	Wilson	2/3	0/3
ChronoBERT_{2009}	**Clinton**	Clinton	**Obama**	Obama	Obama	Wilson	2/3	0/3
ChronoBERT_{2010}	**Clinton**	Obama	**Obama**	Obama	Obama	Wilson	2/3	0/3
ChronoBERT_{2011}	**Clinton**	Clinton	**Obama**	Obama	Obama	Wilson	2/3	0/3
ChronoBERT_{2012}	Obama	Obama	**Obama**	Obama	Obama	Obama	1/3	0/3
ChronoBERT_{2013}	**Clinton**	Obama	**Obama**	Obama	Obama	Monroe	2/3	0/3
ChronoBERT_{2014}	**Clinton**	**Bush**	**Obama**	Obama	Obama	Monroe	3/3	0/3
ChronoBERT_{2015}	**Clinton**	Clinton	**Obama**	Obama	Obama	Monroe	2/3	0/3
ChronoBERT_{2016}	**Clinton**	**Bush**	**Obama**	Obama	Obama	Obama	3/4	0/2
ChronoBERT_{2017}	**Clinton**	**Bush**	**Obama**	**Trump**	Trump	Monroe	4/4	0/2
ChronoBERT_{2018}	**Clinton**	**Bush**	**Obama**	**Trump**	Trump	Obama	4/4	0/2
ChronoBERT_{2019}	**Clinton**	**Bush**	**Obama**	**Trump**	Trump	Obama	4/4	0/2
ChronoBERT_{2020}	**Clinton**	**Bush**	**Obama**	**Trump**	Trump	**Trump**	4/5	1/1
ChronoBERT_{2021}	**Clinton**	Clinton	**Obama**	**Trump**	**Biden**	Biden	4/5	0/1
ChronoBERT_{2022}	**Clinton**	**Bush**	**Obama**	**Trump**	**Biden**	Biden	5/5	0/1
ChronoBERT_{2023}	**Clinton**	**Bush**	**Obama**	**Trump**	**Biden**	Biden	5/5	0/1
ChronoBERT_{2024}	**Clinton**	**Bush**	**Obama**	**Trump**	**Biden**	Biden	5/6	—
ChronoBERT_{1999} through ChronoBERT_{2024}							68/83	1/73

Table 4 Masked-Token Predictions of U.S. Presidents using ChronoBERT

This table displays ChronoBERT's masked-token predictions in a sentence describing an incoming U.S. president, where the president's name is the target of the prediction. The input prompt is,

> "After the {year} U.S. presidential election, President [MASK] was inaugurated as U.S. President in the year {year+1}."

For each masked token, the model's prediction is the token with the highest probability, chosen deterministically. The gray-shaded area denotes predictions covering the post-knowledge cutoff period. The blue text highlights correct predictions. $\text{ChronoBERT}_{\text{Realtime}}$ refers to the latest model before the election year. For comparison, predictions from BERT (released in 2018) and ModernBERT (released in 2024) are also included.

Took office in $\{\text{year}_{p-1}+1\}$: President $\{\text{name}_{p-1}\}$

Took office in $\{\text{year}_{p}+1\}$: President _____".

Table 4 presents the masked-token predictions using ChronoBERT, and Table 5 presents

	Election year						Accuracy	
	1992	2000	2008	2016	2020	2024	Pre-cutoff	Post-cutoff
Correct output	Bill Clinton	George W (Bush)	Barack Obama	Donald Trump	Joe Biden	Donald Trump		
GPT-2	**Bill Clinton**	Bill Clinton	**Barack Obama**	**Donald Trump**	George W	George W	3/4	0/2
GPT-2 XL	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	James A	James Mattis	4/4	0/2
$\text{ChronoGPT}_{\text{Realtime}}$	—	Bill Clinton	George W	Barack Obama	Donald Trump	Joe Biden	—	0/5
ChronoGPT_{1999}	**Bill Clinton**	Bill Clinton	Bill Clinton	Obama\n	Obama\n	Obama\n	1/1	0/5
ChronoGPT_{2000}	**Bill Clinton**	Bill Clinton	Bill Clinton	Bill Clinton	Bill Clinton	Bill Clinton	1/2	0/4
ChronoGPT_{2001}	George W	**George W**	Bill Clinton	George W	George W	George W	1/2	0/4
ChronoGPT_{2002}	**Bill Clinton**	**George W**	George W	George W	George W	George W	2/2	0/4
ChronoGPT_{2003}	**Bill Clinton**	**George W**	George W	George W	George W	George W	2/2	0/4
ChronoGPT_{2004}	**Bill Clinton**	**George W**	George W	George W	George W	George W	2/2	0/4
ChronoGPT_{2005}	**Bill Clinton**	**George W**	George W	George W	George W	George W	2/2	0/4
ChronoGPT_{2006}	**Bill Clinton**	**George W**	George W	George W	George W	George W	2/2	0/4
ChronoGPT_{2007}	**Bill Clinton**	**George W**	George W	George W	George W	George W	2/2	0/4
ChronoGPT_{2008}	**Bill Clinton**	**George W**	George W	George W	George W	George W	2/3	0/3
ChronoGPT_{2009}	**Bill Clinton**	**George W**	**Barack Obama**	Barack Obama	Barack Obama	Barack Obama	3/3	0/3
ChronoGPT_{2010}	**Bill Clinton**	**George W**	**Barack Obama**	Barack Obama	Barack Obama	Barack Obama	3/3	0/3
ChronoGPT_{2011}	**Bill Clinton**	**George W**	**Barack Obama**	Barack Obama	Barack Obama	Barack Obama	3/3	0/3
ChronoGPT_{2012}	**Bill Clinton**	**George W**	**Barack Obama**	Barack Obama	Barack Obama	Barack Obama	3/3	0/3
ChronoGPT_{2013}	**Bill Clinton**	**George W**	**Barack Obama**	Barack Obama	Barack Obama	Barack Obama	3/3	0/3
ChronoGPT_{2014}	**Bill Clinton**	**George W**	**Barack Obama**	Barack Obama	Barack Obama	Barack Obama	3/3	0/3
ChronoGPT_{2015}	**Bill Clinton**	**George W**	**Barack Obama**	Barack Obama	Barack Obama	Barack Obama	3/3	0/3
ChronoGPT_{2016}	**Bill Clinton**	**George W**	**Barack Obama**	Barack Obama	Barack Obama	Barack Obama	3/4	0/2
ChronoGPT_{2017}	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	Donald Trump	**Donald Trump**	4/4	1/2
ChronoGPT_{2018}	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	Donald Trump	**Donald Trump**	4/4	1/2
ChronoGPT_{2019}	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	Donald Trump	**Donald Trump**	4/4	1/2
ChronoGPT_{2020}	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	Donald Trump	**Donald Trump**	4/5	1/1
ChronoGPT_{2021}	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	**Joe Biden**	Joe Biden	5/5	0/1
ChronoGPT_{2022}	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	**Joe Biden**	Joe Biden	5/5	0/1
ChronoGPT_{2023}	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	**Joe Biden**	Joe Biden	5/5	0/1
ChronoGPT_{2024}	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	**Joe Biden**	Joe Biden	5/6	—
ChronoGPT_{1999} through ChronoGPT_{2024}							77/83	4/73

Table 5 Next-Token Predictions of U.S. Presidents using ChronoGPT

This table displays ChronoGPT's next-token predictions for a list including the incoming U.S. president and his three most recent predecessors, where the name of the most recent president is the target of the prediction. The input prompt is,

> "U.S. Presidents in chronological order:
> Took office in $\{\text{year}_{p-3}+1\}$: President $\{\text{name}_{p-3}\}$
> Took office in $\{\text{year}_{p-2}+1\}$: President $\{\text{name}_{p-2}\}$
> Took office in $\{\text{year}_{p-1}+1\}$: President $\{\text{name}_{p-1}\}$
> Took office in $\{\text{year}_{p}+1\}$: President _____".

Each sequence includes exactly two predicted tokens, with the most probable token selected deterministically at each step. The gray-shaded area denotes predictions covering the post-knowledge cutoff period, including election years where the president-elect has yet to be inaugurated. The blue text highlights correct predictions. $\text{ChronoGPT}_{\text{Realtime}}$ refers to the latest model before the election year. For comparison, predictions from GPT-2 and GPT-2 XL (released in 2019) are also included.

the next-token predictions using ChronoGPT. In both tables, the gray-shaded area in the top-right indicates predictions in the post-knowledge cutoff period, and the non-shaded lower-right denotes predictions strictly in the pre-knowledge cutoff period. Correct predictions are highlighted in blue.

If a model is well-trained with high-quality and time-relevant textual data, it should

be able to identify presidents taking office before its knowledge cutoff with a high degree of accuracy. At the same time, an absence of leakage would be evidenced by the model's inability to predict any president elected for the first time after that cutoff, confirming that no future information was inadvertently included in the training data. In other words, for a well-trained model that is perfectly chronologically consistent, we would expect most predictions in the non-shaded area to appear in blue text, while all predictions in the gray-shaded area should appear in black text.

From the results for U.S. presidents, we see that in the pre-cutoff period represented by the non-shaded area, ChronoBERT correctly makes a majority of predictions (68 out of 83), outperforming the original BERT model (2 out of 4), though falling short of ModernBERT's perfect performance. ChronoGPT also correctly makes most predictions (77 out of 83), outperforming GPT-2 (3 out of 4), and coming closer to GPT-2 XL's perfect performance. In contrast, during the post-cutoff period represented by the gray-shaded area, none of the ChronoBERT or ChronoGPT models correctly predict a future president in his first term. Notably, one ChronoBERT model (ChronoBERT_{2020}) and four ChronoGPT models (ChronoGPT_{2017} through ChronoGPT_{2020}) correctly predict President Trump's second non-consecutive term in 2025—this reflects a tendency to favor either a recently elected or historically notable president in an out-of-knowledge context. For ChronoGPT in particular, this also reflects a tendency toward autoregressive repetition, as President Trump is the latest president provided in the input sequence. Therefore, none of the ChronoBERT or ChronoGPT models demonstrates foreknowledge of a future first-term president.

In addition to identifying the names of presidents, we consider a test that involves identifying significant events across different years. For each event, we construct a sentence with a missing token that represents a defining element of the event and is the target of the prediction. The sentences cover the Enron *scandal*, the *SARS* outbreak, the subprime mortgage *crisis*, the *Brexit* referendum, the *coronavirus*/*COVID* pandemic, and

Panel A: Input prompts

Prompt year	Input prompt	Correct output
2001	In 2001, the Enron [MASK] precipitated the passage of the Sarbanes-Oxley Act of 2002.	scandal
2003	In 2003, the [MASK] outbreak led to a major public health crisis.	SARS
2008	In 2008, the subprime mortgage [MASK] severely impacted the global economy.	crisis
2016	In 2016, the uncertainty surrounding the [MASK] referendum led to heightened market volatility.	Brexit/EU
2020	In 2020, the [MASK] pandemic severely impacted the global economy.	coronavirus
2022	In 2022, the public release of [MASK]GPT marked a major milestone for generative AI.	Chat

Panel B: Output predictions

	Event year						Accuracy	
	2001	2003	2008	2016	2020	2022	Pre-cutoff	Post-cutoff
Correct output	scandal	SARS	crisis	Brexit/EU	coronavirus	Chat		
BERT	**scandal**	flu	**crisis**	**EU**	flu	the	3/4	0/2
ModernBERT	**scandal**	**SARS**	**crisis**	**Brexit**	**coronavirus**	Auto	5/6	—
$\text{ChronoBERT}_{\text{Realtime}}$	War	Ebola	program	presidential	influenza	X	—	0/6
ChronoBERT_{1999}	War	influenza	tax	presidential	influenza		—	0/6
ChronoBERT_{2000}	War	Ebola	market	presidential	influenza	O	—	0/6
ChronoBERT_{2001}	War	Ebola	market	presidential	influenza	p	0/1	0/5
ChronoBERT_{2002}	Amendment	Ebola	rate	election	influenza	EX	0/1	0/5
ChronoBERT_{2003}	**scandal**	influenza	rate	Palestinian	influenza	E	1/2	0/4
ChronoBERT_{2004}	**scandal**	**SARS**	rate	tax	influenza	SU	2/2	0/4
ChronoBERT_{2005}	**scandal**	**SARS**	rate	presidential	influenza	O	2/2	0/4
ChronoBERT_{2006}	**scandal**	**SARS**	credit	presidential	influenza	O	2/2	0/4
ChronoBERT_{2007}	**scandal**	**SARS**	program	presidential	influenza	AR	2/2	0/4
ChronoBERT_{2008}	**scandal**	**SARS**	**crisis**	Kyoto	influenza	AR	3/3	0/3
ChronoBERT_{2009}	**scandal**	**SARS**	**crisis**	Kyoto	influenza	E	3/3	0/3
ChronoBERT_{2010}	**scandal**	**SARS**	**crisis**	tax	influenza	AR	3/3	0/3
ChronoBERT_{2011}	**scandal**	**SARS**	**crisis**	Iraq	influenza	U	3/3	0/3
ChronoBERT_{2012}	**scandal**	**SARS**	**crisis**	presidential	influenza	AR	3/3	0/3
ChronoBERT_{2013}	**scandal**	**SARS**	**crisis**	voter	influenza	e	3/3	0/3
ChronoBERT_{2014}	**scandal**	**SARS**	**crisis**	Iraq	influenza	i	3/3	0/3
ChronoBERT_{2015}	**scandal**	**SARS**	**crisis**	presidential	Ebola	i	3/3	0/3
ChronoBERT_{2016}	**scandal**	**SARS**	**crisis**	**Brexit**	influenza	i	4/4	0/2
ChronoBERT_{2017}	**scandal**	Ebola	**crisis**	**Brexit**	Ebola	X	3/4	0/2
ChronoBERT_{2018}	**scandal**	Ebola	**crisis**	**Brexit**	Ebola	X	3/4	0/2
ChronoBERT_{2019}	**scandal**	**SARS**	**crisis**	**Brexit**	influenza	W	4/4	0/2
ChronoBERT_{2020}	**scandal**	**SARS**	**crisis**	**Brexit**	**coronavirus**	V	5/5	0/1
ChronoBERT_{2021}	**scandal**	Ebola	**crisis**	**Brexit**	**coronavirus**	X	4/5	0/1
ChronoBERT_{2022}	**scandal**	Ebola	**crisis**	**Brexit**	**coronavirus**	#	4/6	—
ChronoBERT_{2023}	**scandal**	**SARS**	**crisis**	**Brexit**	**coronavirus**	**Chat**	6/6	—
ChronoBERT_{2024}	**scandal**	**SARS**	**crisis**	**Brexit**	**coronavirus**	**Chat**	6/6	—
ChronoBERT_{1999} through ChronoBERT_{2024}							72/80	0/76

Table 6 Masked-Token Predictions of Major Events using ChronoBERT

This table reports ChronoBERT's masked-token predictions for sentences describing major events from various years. Panel A shows the input prompts for each event. Panel B shows the output predictions for each event by each model in the ChronoBERT series. For each masked token, the model's prediction is the token with the highest probability, chosen deterministically. The gray-shaded area denotes predictions covering the post-knowledge cutoff period. The blue text highlights correct predictions. $\text{ChronoBERT}_{\text{Realtime}}$ refers to the latest model before the event year. For comparison, predictions from BERT (released in 2018) and ModernBERT (released in 2024) are also included.

the public release of *ChatGPT*.

Table 6 presents the masked-token predictions using ChronoBERT, and Table 7 presents the next-token predictions using ChronoGPT. The specific textual sequences containing the event-defining missing tokens are reported in each table. In the pre-cutoff period represented by the non-shaded area, ChronoBERT correctly makes a majority of predic-

Panel A: Input prompts

Prompt year	Input prompt	Correct output
2001	The Sarbanes-Oxley Act was introduced in response to the 2001 Enron _____	scandal
2003	In 2003, a major public health crisis was the outbreak of the virus known as _____	SARS
2008	In 2008, the global economy was dominated by the subprime mortgage _____	crisis
2016	In 2016, market volatility increased due to the uncertainty surrounding the vote in the UK, known as the _____	Brexit referendum/vote
2020	In 2020, the global economy was severely impacted by the pandemic known as _____	COVID
2022	In 2022, a major milestone for generative AI was marked by the public release of the AI assistant known as _____	ChatGPT

Panel B: Output predictions

	Prompt year						Accuracy	
	2001	2003	2008	2016	2020	2022	Pre-cutoff	Post-cutoff
Correct output	scandal	SARS	crisis	Brexit referendum/vote	COVID	ChatGPT		
GPT-2	**scandal**.\n	chikung	**crisis**, which	**Brexit vote**.	the Ebola virus	the "AI	3/4	0/2
GPT-2 XL	**scandal**, in	**SARS**,	**crisis**. The	**Brexit**, and	the "Great	Alexa. Alexa	4/4	0/2
$\text{ChronoGPT}_{\text{Realtime}}$	a-H	hantav	market, which	"UK vote	the Ebola virus	Alexa. The	—	0/6
ChronoGPT_{1999}	ous Claims Act	the Ebola virus	market. The	"Budget	the "green	the AI assistant	—	0/6
ChronoGPT_{2000}	a-H	Ebola virus in	rate, which	"B	the Ebola virus	AI-R	—	0/6
ChronoGPT_{2001}	Agreement, which	Ebola (E	market	"V	the Ebola virus	AI Magazine.	0/1	0/5
ChronoGPT_{2002}	**scandal**. The	hantav	market, which	"V	the "Asian	the AI-R	1/1	0/5
ChronoGPT_{2003}	**scandal**. It	West Nile virus	market, which	"Voting	the Ebola virus	AIBO.	1/2	0/4
ChronoGPT_{2004}	**scandal**. It	**SARS**.	market, which	"Voting	the SARS	AIBO.	2/2	0/4
ChronoGPT_{2005}	**scandal**. It	**SARS** (	market, which	"London effect	SARS.	AIBO.	2/2	0/4
ChronoGPT_{2006}	**scandal**. It	**SARS**,	market, which	"London effect	the SARS	the AIBO	2/2	0/4
ChronoGPT_{2007}	**scandal**. It	**SARS**,	market, which	"Budget	the Black Death	the AIBO	2/2	0/4
ChronoGPT_{2008}	**scandal**. It	**SARS**,	**crisis**, which	"Voting	the Black Death	the AIBO	3/3	0/3
ChronoGPT_{2009}	**scandal**. The	**SARS**,	**crisis**, which	"UK vote	the Black Death	the "C	3/3	0/3
ChronoGPT_{2010}	**scandal**. The	**SARS**,	**crisis**, which	"UK vote	the SARS	the AIBO	3/3	0/3
ChronoGPT_{2011}	**scandal**. The	**SARS**,	**crisis**, which	"Budget	the H1	the AIBO	3/3	0/3
ChronoGPT_{2012}	**scandal**. The	**SARS**,	**crisis**, which	"Budget	the Black Death	the AI-	3/3	0/3
ChronoGPT_{2013}	**scandal**. The	**SARS**,	**crisis**, which	"Budget	the "Asian	the AIBO	3/3	0/3
ChronoGPT_{2014}	**scandal**. The	**SARS**,	**crisis**, which	"Voting	the "Asian	the "C	3/3	0/3
ChronoGPT_{2015}	**scandal**. The	**SARS**,	**crisis**, which	"UK vote	the Ebola virus	the AIBO	3/3	0/3
ChronoGPT_{2016}	**scandal**. The	**SARS**,	**crisis**, which	"UK vote	the Ebola virus	the "C	3/4	0/2
ChronoGPT_{2017}	**scandal**. The	**SARS**,	**crisis**, which	**Brexit referendum**.	the Ebola virus	the "C	4/4	0/2
ChronoGPT_{2018}	**scandal**. The	Ebola in West	**crisis**, which	**Brexit vote**.	the Ebola virus	DeepMind's	3/4	0/2
ChronoGPT_{2019}	**scandal**. The	Ebola in West	**crisis**, which	**Brexit referendum**.	the Ebola virus	Siri. The	3/4	0/2
ChronoGPT_{2020}	**scandal**. The	**SARS-**	**crisis**, which	**Brexit referendum**.	**COVID-**	Siri. The	5/5	0/1
ChronoGPT_{2021}	**scandal**. The	**SARS-**	**crisis**, which	**Brexit referendum**.	**COVID-**	Alexa. The	5/5	0/1
ChronoGPT_{2022}	**scandal**. The	**SARS-**	**crisis**, which	**Brexit referendum**.	**COVID-**	Alexa. The	5/6	—
ChronoGPT_{2023}	**scandal**. The	**SARS-**	**crisis**, which	**Brexit referendum**.	**COVID-**	**ChatGPT**	6/6	—
ChronoGPT_{2024}	**scandal**. The	**SARS-**	**crisis**, which	**Brexit referendum**.	**COVID-**	**ChatGPT**	6/6	—
ChronoGPT_{1999} through ChronoGPT_{2024}							74/80	0/76

Table 7 Next-Token Predictions of Major Events using ChronoGPT

This table reports ChronoGPT's next-token predictions for sentences describing major events from various years. Panel A shows the input prompts for each event. Panel B shows the output predictions for each event by each model in the ChronoGPT series. Each sequence includes exactly three predicted tokens, with the most probable token selected deterministically at each step. The gray-shaded area denotes predictions covering the post-knowledge cutoff period. The blue text highlights correct predictions. $\text{ChronoGPT}_{\text{Realtime}}$ refers to the latest model before the event year. For comparison, predictions from GPT-2 and GPT-2 XL (released in 2019) are also included.

tions (72 out of 80), outperforming BERT (3 out of 4) and ModernBERT (5 out of 6). ChronoGPT also correctly makes most predictions (74 out of 80), outperforming GPT-2 (3 out of 4), though slightly below GPT-2 XL's perfect performance. None of the ChronoBERT and ChronoGPT models correctly predicts an event-defining token from a future event, as represented by the gray-shaded area. Overall, these findings show no evidence of leakage in the textual data used to train our chronologically consistent models.

Notably, none of the predictions are correct for ChronoBERT$_{\text{Realtime}}$ and ChronoGPT$_{\text{Realtime}}$, where each model only contains information available up to the year preceding the election or event. This stands in sharp contrast to the high accuracy achieved by the ChronoBERT and ChronoGPT models evaluated on past presidential elections and events within their respective knowledge windows. In this setting—where accurate predictions hinge on specific knowledge of presidents or events—lookahead bias provides the language models with an insurmountable advantage over their real-time counterparts.

3.3.2 Probability of Neologisms

The validation exercises above rely on predicting specific, high-profile tokens (presidential names or major event keywords) and provide evidence that our models do not suffer from lookahead bias regarding major, date-specific facts. To test chronological consistency more systematically, we conduct a broader test based on neologisms—new words and phrases that entered the language at known dates. Rather than checking whether a model predicts a single correct token, this test examines the probability each model assigns to a large set of neologisms, offering a more granular and comprehensive assessment of chronological consistency.

ChronoGPT, as an autoregressive transformer, assigns a probability distribution over all possible next tokens given a prompt. If a model has encountered a word in its training data, it should assign a high probability to the corresponding token sequence in an appropriate context. Conversely, if the model has never encountered the word, the probability should be indistinguishable from zero. If each ChronoGPT vintage is truly representative of its time period, we should observe a sharp increase in the probability it assigns to a neologism precisely at the vintage corresponding to the word's first known use date, with near-zero probability in all earlier vintages.

Using the Time Traveler tool on Merriam-Webster's website, we obtained a list of 395

neologisms, each paired with its first known use date, spanning 2002 to 2022. For each neologism, we used ChatGPT to generate a context-setting sentence of approximately 20 words that ends immediately before the target word. For example, for the phrase "gig economy" (first known use: 2009), the prompt is: "With the rise of apps for ride-sharing and food delivery, more people began participating in the ..."

We feed each prompt to every ChronoGPT vintage and calculate the probability of the target phrase. Since many phrases consist of multiple tokens, the phrase probability is the product of the conditional probabilities of each token in the sequence:

$$P(\text{phrase}|\text{context}) = \prod_{i=1}^{k} P(w_i|\text{context}, w_1, \ldots, w_{i-1})$$

where $w_1, \ldots, w_k$ are the k tokens comprising the target phrase.

To aggregate results, we align all 395 neologisms by their first known use date, denoted $t = 0$. The model vintage is represented in relative time as $t = T - Y_w$, where T is the vintage year and Y_w is the first known use date. We define the increase in phrase probability for neologism w at relative time t as:

$$\Delta P_w(t) = P_{w,Y_w+t} - \frac{1}{N_w} \sum_{k=1}^{N_w} P_{w,Y_w-k}$$

where $P_{w,T}$ is the phrase probability from ChronoGPT vintage T, and $N_w = \min(10, Y_w - 1999)$ is the number of baseline years.

Figure 4 plots the average $\Delta P(t)$ across all 395 neologisms.

Before the first known use date ($t < 0$), the average increase in phrase probability demonstrates no significant deviation from zero, indicating that the models have no prior knowledge of these terms and confirming the absence of leakage in our training corpora.

At $t = 0$, the phrase probability jumps sharply from 0% to 0.3%, confirming that the models begin to recognize new terms as soon as they appear in the training data.

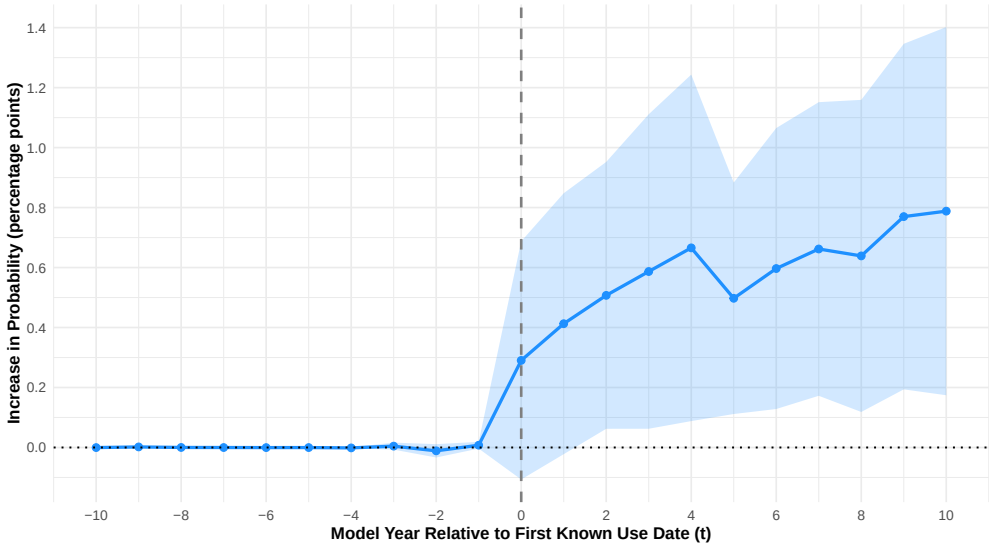


Figure 4 Near-Zero Probability Prior to First Known Use Date

This figure shows the average increase in phrase probability relative to the pre-introduction baseline. The x-axis represents the model vintage year minus the word's introduction year. The shaded area represents the 95% confidence interval.

For $t > 0$, the phrase probability continues to increase gradually, reaching approximately 0.8% by $t + 10$. This gradual rise is expected: a word's first known use date typically precedes its widespread adoption, so it takes time for a neologism to appear frequently enough in the training data for the model to assign it a high probability. The 95% confidence interval (shaded area) is wide, reflecting considerable variance across individual neologisms. Nonetheless, the increase in phrase probability is statistically significant from $t \geq 2$.

This test is not directly transferable to ChronoBERT. As a masked language model, ChronoBERT predicts a single masked token from bidirectional context, and there is no straightforward method to elicit a multi-token sequence probability comparable to the autoregressive calculation used here. However, since ChronoBERT and ChronoGPT are trained on the same chronologically ordered corpora, the absence of leakage in ChronoGPT provides evidence for the temporal integrity of the underlying data, and therefore of both model families.

3.4 Predicting Stock Returns using Financial News

	ChronoBERT$_{\text{Realtime}}$			ChronoGPT$_{\text{Realtime}}$			Llama 3.1			GPT-2 XL		
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
Low (L)	-23.30	25.86	-0.90	-19.82	25.15	-0.79	-23.71	26.15	-0.91	-22.25	25.75	-0.86
2	-2.43	25.20	-0.10	1.48	25.48	0.06	-4.77	25.31	-0.19	-1.36	25.32	-0.05
3	4.17	25.64	0.16	0.61	24.87	0.02	-0.24	24.86	-0.01	-1.10	24.82	-0.04
4	4.17	24.58	0.17	2.41	24.26	0.10	3.84	24.62	0.16	2.60	24.43	0.11
5	3.94	24.22	0.16	7.18	24.54	0.29	7.47	24.65	0.30	6.24	24.30	0.26
6	10.81	24.13	0.45	5.78	24.20	0.24	12.03	24.23	0.50	10.14	24.53	0.41
7	14.56	24.23	0.60	10.82	23.84	0.45	13.31	24.33	0.55	16.90	24.76	0.68
8	16.38	23.64	0.69	15.74	24.30	0.65	15.13	23.79	0.64	15.69	24.09	0.65
9	23.95	24.45	0.98	22.63	24.19	0.94	24.68	23.88	1.03	24.85	24.24	1.03
High (H)	37.71	24.53	1.54	43.08	25.41	1.70	42.20	25.05	1.68	38.25	24.53	1.56
H-L	61.02	12.72	4.80	62.90	12.78	4.92	65.91	13.46	4.90	60.51	13.24	4.57
	ModernBERT			BERT			FinBERT			StoriesLM		
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
Low (L)	-24.39	26.07	-0.94	-22.52	26.21	-0.86	-23.96	26.86	-0.89	-17.80	26.52	-0.67
2	0.49	25.86	0.02	-5.05	25.55	-0.20	-3.17	25.64	-0.12	-1.19	25.26	-0.05
3	1.64	25.08	0.07	3.12	24.92	0.13	3.36	24.83	0.14	1.86	24.92	0.07
4	5.41	24.81	0.22	8.14	24.62	0.33	7.19	24.52	0.29	5.90	24.62	0.24
5	8.67	24.67	0.35	10.81	24.44	0.44	9.17	24.39	0.38	4.99	24.30	0.21
6	10.02	23.99	0.42	9.38	24.02	0.39	11.47	24.03	0.48	11.88	23.90	0.50
7	12.99	23.93	0.54	14.54	23.83	0.61	16.54	23.92	0.69	12.41	23.66	0.52
8	16.17	23.69	0.68	18.51	24.04	0.77	19.16	23.65	0.81	18.93	24.19	0.78
9	23.45	23.97	0.98	19.68	23.90	0.82	20.70	23.88	0.87	23.25	24.30	0.96
High (H)	35.53	24.42	1.45	33.37	24.88	1.34	29.51	24.60	1.20	29.73	24.78	1.20
H-L	59.92	13.03	4.60	55.89	13.38	4.18	53.47	13.85	3.86	47.53	13.90	3.42

Table 8 Performance of the LLM Portfolios

This table presents annualized performance metrics (mean return, standard deviation, and Sharpe ratio) for decile portfolios sorted by next-day return predictions from financial news. Portfolios are rebalanced daily, with the "H-L" row representing a strategy of longing the top decile and shorting the bottom decile. All values are in percentage points except Sharpe ratios. All portfolios are equal-weighted. Data spans January 2008–July 2023.

To gauge lookahead bias in a concrete prediction task, we forecast stock returns using news embeddings from different language models. We construct portfolios by sorting stocks based on each model's return forecasts and evaluate the performance of long-short spreads generated by these rankings.[23]

Table 8 presents the decile portfolio performance for real-time models (ChronoBERT$_{\text{Realtime}}$ and ChronoGPT$_{\text{Realtime}}$). For example, in the year t, we would use the latest available

[23]Following He et al. (2024), we conduct a robustness check by forecasting the probability of a positive stock return on the subsequent trading day. The outcomes closely mirror those obtained from the return forecasts.

model checkpoint at year $t-1$ to embed the news articles in that year and make predictions based on the embeddings. For comparison, we also report six other benchmarks: (1) Llama-3.1-8B; (2) GPT-2 XL; (3) ModernBERT; (4) BERT; (5) FinBERT; and (6) StoriesLM. In this news return prediction setting, the H-L portfolios from ChronoBERT$_{\text{Realtime}}$ and ChronoGPT$_{\text{Realtime}}$ generate Sharpe ratios of 4.80 and 4.92, outperforming GPT-2 XL, ModernBERT, BERT, FinBERT, and StoriesLM.

	ChronoBERT	ChronoGPT	Llama 3.1	GPT-2 XL	ModernBERT	BERT	FinBERT	StoriesLM
ChronoBERT		0.717	0.685	0.141	0.144	0.005	0.002	0.000
ChronoGPT	0.283		0.453	0.051	0.075	0.001	0.001	0.000
Llama 3.1	0.315	0.547		0.048	0.069	0.001	0.000	0.000
GPT-2 XL	0.859	0.950	0.952		0.555	0.038	0.010	0.000
ModernBERT	0.856	0.925	0.931	0.445		0.029	0.007	0.000
BERT	0.995	0.999	0.999	0.962	0.971		0.116	0.005
FinBERT	0.998	0.999	1.000	0.991	0.993	0.884		0.098
StoriesLM	1.000	1.000	1.000	1.000	1.000	0.995	0.902	

Table 9 P-value of Pairwise Sharpe Ratio Difference Tests

This table reports the p-value from the Ledoit and Wolf (2008) Sharpe ratio difference test of the 'H-L' portfolios from different LLMs in Table 8. Each entry corresponds to a test of the null hypothesis that the Sharpe ratio of the model in the row is smaller than or equal to that of the model in the column. The portfolio sample spans from January 2008 to July 2023.

We also report the p-value of the pairwise Sharpe ratio difference test using the Ledoit and Wolf (2008) approach in Table 9. Comparing real-time ChronoBERT and ChronoGPT against BERT, FinBERT, and StoriesLM, we find that the models' improved language understanding and superior knowledge increase the investment Sharpe ratio, a difference that is both economically meaningful and statistically significant at the 1% level.

Comparing the investment performance of ChronoBERT and ChronoGPT against the Llama 3.1 model, we find that ChronoBERT and ChronoGPT generate comparable Sharpe ratios, with no statistically significant differences (p-value of 0.315 and 0.547). Between the two chronological models themselves, we observe no statistical differences (p-value of 0.283), indicating that both approaches effectively leverage temporal information.

These findings suggest that the performance of our chronologically consistent models

is comparable to powerful, larger-scale LLMs in this financial application, despite their limited training using historical textual data.

A somewhat surprising finding is that lookahead bias appears to be modest in this stock return forecasting application. The fact that the Llama-based performance is no different from that of our chronologically consistent models suggests understanding the news flow into the market in real-time is sufficient for generating substantial short-term returns, and that no disruption of the time continuum is required.

Because of the strong factor structure in stock returns, lookahead bias can arise not only because of firm-specific (idiosyncratic) returns, but also because of future knowledge of common factors. Chronologically inconsistent models (e.g., Llama) may "memorize" that a class of stocks performs well over a future interval, rather than learning a firm's own price path. For instance, if small-cap stocks rally in a later period, a model with future leakage can encode that pattern and load on the size factor.

To test for lookahead bias in common factors, we report alphas relative to four benchmark specifications: the Fama and French (2015) five-factor model (α_{F5}), Fama and French (2018) six-factor model (α_{F6}), the Hou et al. (2015) q-factor model (α_{HXZ}), and the Daniel et al. (2020) behavioral factor model (α_{DHS}). If alphas persist after these adjustments, there is a modest bias attributed to standard common factors. If they attenuate, the factor "memorization" is a potential channel for lookahead bias.

Table 10 compares the long-short portfolio performances across models. $\text{ChronoGPT}_{\text{Realtime}}$ and $\text{ChronoBERT}_{\text{Realtime}}$ earn mean annualized returns of about 63 and 61 percentage points, matching or exceeding most baselines. Crucially, subtracting standard factor premia hardly moves the estimates. For ChronoBERT, the alpha is between 60.10 and 60.71 percentage points. For ChronoGPT, it ranges from 62.73 to 62.96 percentage points. The associated t-statistics remain virtually unchanged. Llama 3.1 shows a similarly small drop when factors are included (from 65.91 to 65.01–65.82 percentage points).

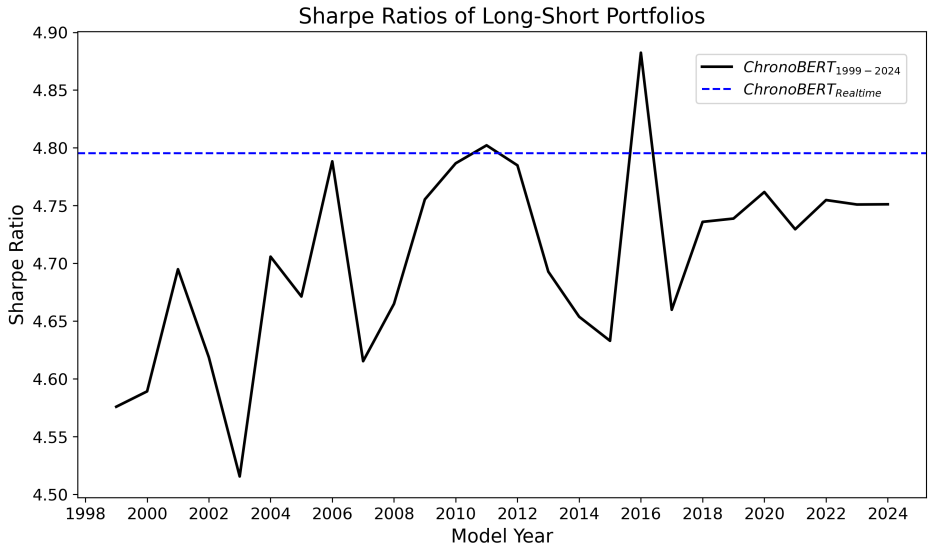
	Mean	α_{F5}	α_{F6}	α_{HXZ}	α_{DHS}
$\text{ChronoBERT}_{\text{Realtime}}$	61.02	60.50	60.55	60.71	60.10
	(4.59)	(4.57)	(4.58)	(4.54)	(4.59)
$\text{ChronoGPT}_{\text{Realtime}}$	62.90	62.87	62.89	62.96	62.73
	(4.53)	(4.52)	(4.52)	(4.44)	(4.49)
Llama 3.1	65.91	65.01	65.06	65.82	64.95
	(4.75)	(4.68)	(4.69)	(4.68)	(4.74)
GPT-2 XL	60.51	59.48	59.52	60.37	59.36
	(4.45)	(4.36)	(4.36)	(4.38)	(4.43)
ModernBERT	59.92	59.41	59.49	59.24	59.05
	(4.42)	(4.34)	(4.35)	(4.32)	(4.38)
BERT	55.89	54.36	54.50	55.37	54.72
	(4.28)	(4.20)	(4.23)	(4.22)	(4.26)
FinBERT	53.47	52.11	52.27	53.54	52.81
	(4.23)	(4.19)	(4.20)	(4.10)	(4.15)
StoriesLM	47.53	46.56	46.67	47.16	46.28
	(4.29)	(4.24)	(4.26)	(4.25)	(4.26)

Table 10 Portfolio Alphas

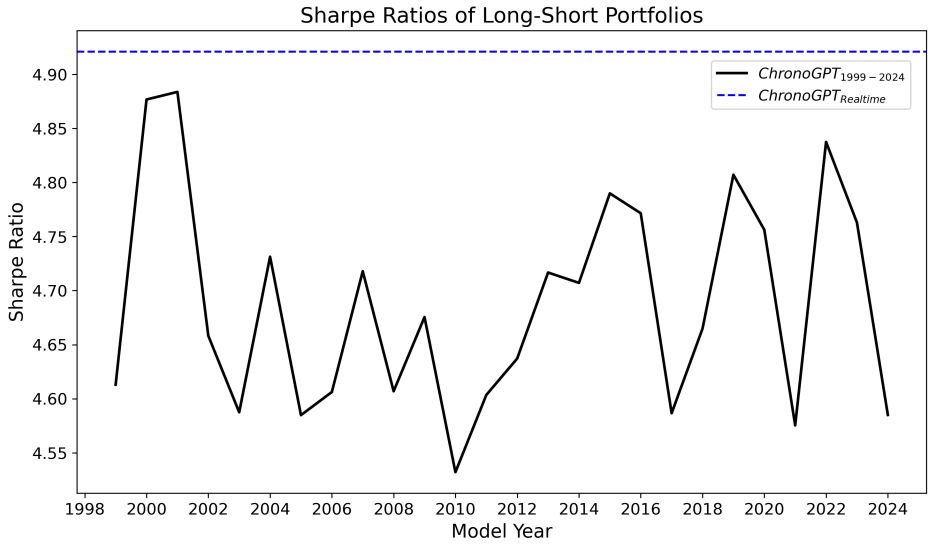
This table presents the portfolio alpha of different models. We report the alphas of long-short portfolios relative to four factor models: Fama and French (2015) five factors (α_{F5}), Fama and French (2018) six factors (α_{F6}), Hou et al. (2015) factors (α_{HXZ}), and Daniel et al. (2020) factors (α_{DHS}). We report Newey-West standard errors with 21 lags in parentheses. The portfolio sample spans from January 2008 to July 2023.

These minimal adjustments imply that common-factor comovement explains little of the reported predictability. The signal is largely orthogonal to standard risk factors. Hence, the lookahead bias related to factor "memorization" for the strategy is also modest.

One may wonder if the return predictability from news text represents investors' underreaction or overreaction to information. In Figure D.1, we track the performance of long-short portfolios from Day 0 (the trading day contemporaneous with the news release) through Day 6. We find significant positive returns on Day 0, followed by a fast decay through Day 2. The evidence is consistent with investors underreacting to textual information in news, and the information is rapidly incorporated into stock prices.



(a) ChronoBERT



(b) ChronoGPT

Figure 5 Importance of Up-to-date Knowledge

This figure illustrates the Sharpe ratios of long-short portfolios constructed using predictions derived from financial news, with language models pretrained on text data up to the time points indicated on the x-axis. The blue dashed line represents the performance of the chronologically consistent real-time models.

3.5 The Superiority of Temporal Alignment

While our models demonstrate more up-to-date knowledge and improved language understanding as they are trained with more data over time (Figure 3), these improvements may or may not translate into economic gains. To test this, we evaluate the trading performance of the entire series of chronologically consistent models. If up-to-date knowledge is the bottleneck of performance in return prediction tasks, we would expect to see a monotonic increase in the Sharpe ratio as the models' knowledge cutoff dates advance.

Figure 5 reveals a surprisingly different outcome. For each time t, we use the corresponding model from that vintage to generate predictions from news embeddings. Instead of a steady increase, we observe a distinct "envelope" phenomenon across both ChronoBERT and ChronoGPT vintages. The real-time model's Sharpe ratio outperforms most vintage models, even those with more up-to-date knowledge cutoff.

These results suggest two immediate conclusions. First, lookahead bias is modest. If significant lookahead bias is present, the final 2024 model, with the most comprehensive knowledge, would have been the top performer across all periods. Second, further improvements in knowledge and generic language ability add only marginal value for this task: the earliest model is already close to the performance frontier.

The envelope pattern suggests that the temporal alignment of language models results in better return prediction. However, the exact mechanism of this alignment is not immediately obvious. We consider three potential channels through which temporal alignment could help:

1. **Language Understanding:** The real-time model might simply possess a superior grasp of the specific linguistic structures used in its own period.

2. **Catastrophic Forgetting:** Later vintages might suffer from degraded performance

	Corpus year																
	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
ChronoGPT_{1999}	29.44	30.07	28.00	27.43	27.28	29.39	27.55	26.22	24.90	24.27	24.38	24.04	24.59	24.96	25.49	26.42	27.88
ChronoGPT_{2000}	26.74	27.30	25.66	25.14	24.97	26.87	25.19	24.05	22.77	22.17	22.19	21.81	22.21	22.55	23.11	23.91	25.27
ChronoGPT_{2001}	25.39	25.96	24.55	24.05	23.90	25.48	24.15	23.06	21.89	21.36	21.41	21.03	21.43	21.75	22.22	23.04	24.37
ChronoGPT_{2002}	24.62	25.10	23.82	23.33	23.14	24.59	23.41	22.36	21.18	20.62	20.65	20.23	20.61	20.92	21.39	22.21	23.54
ChronoGPT_{2003}	22.86	23.30	22.08	21.62	21.49	22.64	21.65	20.69	19.68	19.22	19.26	18.87	19.17	19.49	19.92	20.64	21.82
ChronoGPT_{2004}	21.54	21.97	20.74	20.27	20.10	21.05	20.31	19.42	18.52	18.15	18.19	17.85	18.09	18.42	18.79	19.41	20.50
ChronoGPT_{2005}	20.05	20.42	19.33	18.90	18.78	19.53	18.94	18.15	17.36	17.05	17.06	16.74	16.95	17.28	17.62	18.20	19.18
ChronoGPT_{2006}	**19.36**	19.66	18.72	18.29	18.15	18.83	18.35	17.59	16.84	16.55	16.58	16.27	16.43	16.76	17.04	17.58	18.49
ChronoGPT_{2007}	18.44	**18.70**	17.94	17.55	17.41	18.07	17.60	16.88	16.17	15.90	15.94	15.63	15.78	16.13	16.40	16.91	17.76
ChronoGPT_{2008}	18.09	18.14	**17.47**	17.16	17.04	17.62	17.25	16.58	15.87	15.58	15.65	15.36	15.52	15.84	16.09	16.59	17.44
ChronoGPT_{2009}	17.79	17.84	16.98	**16.77**	16.69	17.25	16.85	16.18	15.50	15.23	15.30	15.07	15.26	15.54	15.79	16.25	17.10
ChronoGPT_{2010}	17.68	17.64	16.82	16.43	**16.39**	16.94	16.60	15.94	15.28	15.00	15.07	14.88	15.08	15.35	15.59	16.05	16.90
ChronoGPT_{2011}	17.56	17.49	16.68	16.26	16.08	**16.68**	16.35	15.66	15.00	14.73	14.82	14.63	14.81	15.07	15.30	15.75	16.58
ChronoGPT_{2012}	17.50	17.41	16.61	16.21	15.99	16.39	**16.20**	15.56	14.89	14.62	14.71	14.53	14.72	14.98	15.17	15.62	16.46
ChronoGPT_{2013}	17.36	17.32	16.52	16.11	15.90	16.32	15.95	**15.39**	14.73	14.44	14.55	14.38	14.56	14.83	14.99	15.45	16.29
ChronoGPT_{2014}	16.50	16.48	15.76	15.38	15.20	15.58	15.23	14.67	**14.10**	13.85	13.95	13.78	13.97	14.21	14.37	14.82	15.62
ChronoGPT_{2015}	16.02	16.01	15.29	14.90	14.74	15.06	14.77	14.20	13.60	**13.43**	13.57	13.40	13.55	13.80	13.93	14.35	15.11
ChronoGPT_{2016}	16.09	16.10	15.35	14.94	14.76	15.06	14.80	14.22	13.57	13.35	**13.54**	13.38	13.51	13.76	13.88	14.29	15.04
ChronoGPT_{2017}	15.73	15.76	15.01	14.61	14.44	14.74	14.50	13.95	13.31	13.05	13.18	**13.06**	13.20	13.44	13.57	13.96	14.69
ChronoGPT_{2018}	15.72	15.74	15.03	14.63	14.45	14.80	14.50	13.96	13.30	13.02	13.09	12.93	**13.15**	13.41	13.52	13.90	14.64
ChronoGPT_{2019}	15.80	15.81	15.08	14.66	14.49	14.81	14.54	14.01	13.33	13.03	13.10	12.88	13.02	**13.32**	13.45	13.83	14.58
ChronoGPT_{2020}	15.94	16.00	15.27	14.87	14.69	15.09	14.73	14.18	13.47	13.13	13.19	12.97	13.07	13.27	**13.47**	13.89	14.67
ChronoGPT_{2021}	16.05	16.06	15.35	14.96	14.78	15.25	14.84	14.28	13.55	13.20	13.25	13.01	13.11	13.21	13.37	**13.85**	14.66
ChronoGPT_{2022}	16.21	16.23	15.51	15.15	14.96	15.50	15.02	14.44	13.70	13.31	13.36	13.12	13.24	13.33	13.45	13.86	**14.73**
ChronoGPT_{2023}	15.67	15.68	14.90	14.49	14.30	14.62	14.33	13.80	13.13	12.86	12.92	12.67	12.79	12.88	12.92	13.25	13.93
ChronoGPT_{2024}	16.07	16.03	15.32	14.92	14.73	15.10	14.75	14.17	13.47	13.14	13.19	12.94	13.13	13.24	13.23	13.59	14.32

Table 11 ChronoGPT Perplexity on the Dow Jones Newswires Corpus

This figure reports the perplexity scores of each ChronoGPT model vintage on each year of the Dow Jones Newswires corpus. The numbers in bold correspond to the real-time model. The model with the lowest perplexity is underlined for each corpus year. A darker shade of green indicates lower perplexity.

on historical data due to the forgetting of past patterns during subsequent training.

3. **Contextual Knowledge:** The ability to understand and interpret the news within the specific information environment of the time.

A natural hypothesis is that the real-time model performs better because it possesses a better understanding of the language used at time t. We first revisit general language abilities in Figure 3, where standard benchmarks (HellaSwag and GLUE) show monotonic improvement for ChronoBERT and ChronoGPT over time. The 2024 vintage is objectively a more capable language model than the 2010 vintage. However, static benchmarks do not account for the fact that language evolves. It remains possible that while the 2024 vintage is generally "smarter," it has lost the specific linguistic nuances of 2010 that the 2010 vintage captured perfectly.

Motivated by Loureiro et al. (2022), who evaluate diachronic language models by

computing pseudo-perplexity across time-specific test sets, we compute perplexity scores for each ChronoGPT vintage on each year of the Dow Jones Newswire corpus. Notably, because our models are trained on non-Dow Jones Newswires data, this exercise also tests whether temporal alignment extends across text types—that is, whether models trained on general-purpose corpora exhibit improved language understanding on contemporaneous financial text they have never seen. This produces a model-year $\times$ corpus-year matrix, allowing us to compare any model's language understanding on any year's text.

Table 11 presents the results. A lower perplexity, shaded in darker green, indicates better language understanding of the corpus text. The results suggest that the language understanding mechanism is unlikely to be the driver. For every corpus year from 2007 to 2023, the majority of later model vintages achieve lower perplexity than the real-time model, indicated by numbers in bold. The real-time model is never the language model with the lowest perplexity, yet it is precisely this model that generates the highest trading returns.

The perplexity results in Table 11 also make catastrophic forgetting an unlikely explanation. If later vintages were severely forgetting earlier linguistic patterns during subsequent training, their perplexity on historical text would increase. Instead, reading down any column of the matrix, perplexity overwhelmingly decreases with vintage year. For instance, on 2007 news text, perplexity falls from 18.44 for ChronoGPT$_{2007}$ to 16.07 for ChronoGPT$_{2024}$. Later models do not forget earlier language; they model it with increasing accuracy. Moreover, the event prediction results reported in Section 3.3 show that later vintage models do at least as well as earlier ones on recalling historical facts.

The remaining contextual knowledge channel therefore appears to be the most plausible explanation for the envelope pattern. While later vintages understand the text better (lower perplexity), they are less effective at interpreting the price signal (lower Sharpe ratio). This divergence suggests that effective financial interpretation requires the model

to capture the Zeitgeist—to effectively "live in the moment." A real-time model is more sensitive to the current events and sentiment of its specific era, allowing it to interpret news with the appropriate immediacy and context. Later vintages, despite their superior linguistic abilities, view the same articles through a "sterilized" historical lens. They are less sensitive to the immediate, lived context needed to translate a specific news piece into a price signal as effectively.

3.6 Robustness

While Chrono models are highly effective, they have smaller parameter counts compared to frontier models. Recent evidence from Lopez-Lira et al. (2025) suggests that larger models exhibit stronger memorization than smaller models. On the other hand, Didisheim et al. (2025) show that language models have a greater tendency for lookahead bias with longer-horizon outcomes. In this section, we directly address these dimensions by examining model scaling performance within our controlled architecture and extending our evaluations to longer-horizon returns.

3.6.1 Different Model Sizes

To study the effect of model size on the final outcome, we provide a series of scaling experiments. According to Gadre et al. (2024), the scaling performance on small models is highly predictive of much larger models for the downstream tasks in overtraining regimes.

Therefore, in addition to the baseline results, we train three additional ChronoGPT$_{1999}$ models with sizes of 124M, 353M, and 1B. Combined with the baseline ChronoGPT$_{1999}$ of 1.5B, this represents a series of scaling experiments where model size increases 12 fold. Appendix B presents the model fits, along with the HellaSwag evaluations.

To measure the lookahead bias of these models in embedding-based return prediction

settings, we compare the asset pricing performance of the different sized Chrono models against that of the 1.5B ChronoGPT$_{2024}$ model which approximates the upper bound for performance of smaller-sized ChronoGPT$_{2024}$ models. Specifically, we use the return prediction framework in Section 2.3.2 to measure the decile long-short portfolio Sharpe Ratios from different-sized models.

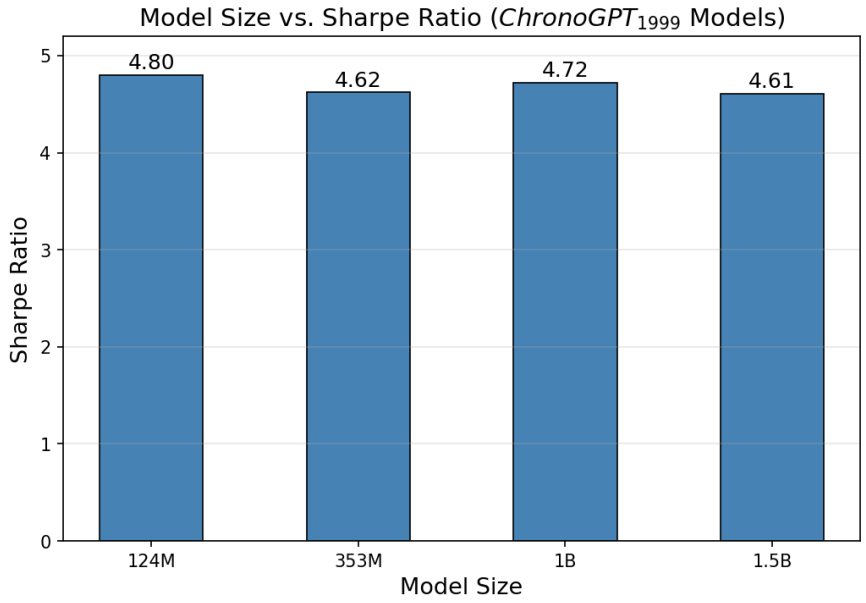


Figure 6 Model Size and Sharpe Ratios (ChronoGPT$_{1999}$ Models)

This figure compares the Sharpe Ratios of ChronoGPT models with four sizes: 124 million (124M), 353 million (353M), 1 billion (1B), and 1.5 billion (1.5B) parameters. All models have a consistent knowledge cutoff of December 1999.

Figure 6 presents the performance of ChronoGPT$_{1999}$ models with four different sizes. Compared to the 1.5B ChronoGPT$_{2024}$ model, which achieves a Sharpe Ratio of 4.58 (Figure 5), all models achieve similar Sharpe ratios. The performance difference between the 1999 model and the 2024 model is almost flat across different sizes. The evidence suggests that lookahead bias is modest across a spectrum of increasing sizes of Chrono models. While we cannot definitively rule out emergent memorization in trillion-parameter frontier models, this evidence suggests that within purpose-built, smaller-scale models, scaling up parameters does not aggressively amplify lookahead bias.

3.6.2 Longer-Horizon Returns

To investigate whether ChronoBERT and ChronoGPT suffer from lookahead bias at extended horizons, we expand our news-based forecasting experiment to 5-, 10-, 21-, and 63-day periods.

Following Jiang et al. (2021), we employ a rolling-window daily turnover strategy: for a k-day horizon, the portfolio on day t is equally weighted across long-short deciles formed from return forecasts over the past k trading days (from $t-1$ to $t-k-1$). On day $t+1$, we rebalance $1/k$ of the portfolio by dropping the oldest position and adding the newest forecast, ensuring consistent exposure across all horizons.

Figure 7 plots the annualized Sharpe ratios across forecasting horizons for ChronoBERT, ChronoGPT, and the benchmark Llama 3.1 model. Several patterns emerge. First, all models exhibit the expected decline in Sharpe ratios as the horizon lengthens, indicating that most news-based predictive content is incorporated into prices within one to two weeks. Second, ChronoBERT and ChronoGPT show strikingly similar performance between their same-year and previous-year vintages across all horizons[24]. This near-overlap confirms that lookahead bias remains minimal, even when extending predictions to the monthly or quarterly scale. Third, Llama 3.1 outperforms the chronologically consistent models for longer horizons, particularly at 21- and 63-day forecasts, suggesting that larger model capacity helps capture slower-moving or cross-sectional fundamentals reflected in longer-horizon price dynamics.

The modest lookahead bias we observe across all horizons is consistent with recent evidence from Didisheim et al. (2025), who find that lookahead bias is most pronounced at lower data frequencies and in aggregate market indexes. Our results extend this understanding to the daily frequency and individual stock level: for high-frequency, stock-

[24]We test the null hypothesis that Sharpe ratios are equal across year and year-1 model specifications using the methodology of Ledoit and Wolf (2008). The differences are insignificant at all horizons ($p > 0.10$).

specific applications, the incremental knowledge from additional training data within the same year adds minimal predictive value.

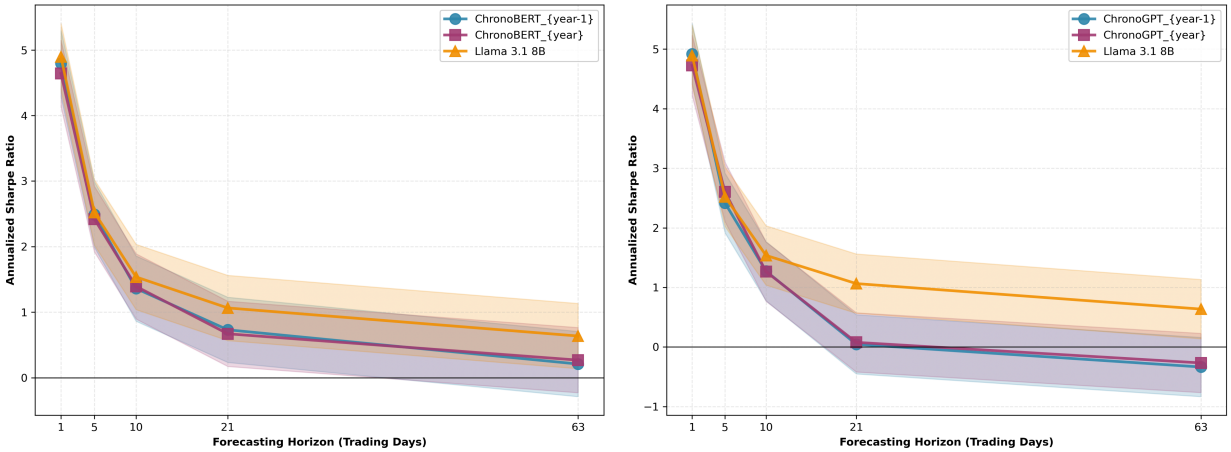


Figure 7 Performance of Longer Horizon Predictions

This figure shows annualized Sharpe ratios for long-short portfolios based on return predictions from ChronoBERT and ChronoGPT over 1-, 5-, 10-, 21-, and 63-day horizons. For each model, we compare two specifications: models where the news year matches the model vintage year versus models where the model vintage is one year behind the news year. At horizons beyond one day, we employ a daily-rebalanced rolling window strategy that maintains k-day exposure by replacing 1/k of the portfolio each day. Shaded regions indicate 95% confidence intervals.

3.7 Portfolio formation, text length, and news coverage intensity

In Appendix D, we provide additional robustness tests. Table D.4 demonstrates that our conclusions remain robust when portfolios are sorted using coarser tercile breakpoints.

The ability of LLMs to use forward information may increase with the length of the sampled text sequence. In Table D.5, we expand the analysis from headlines to full news text and use it to predict future stock returns. Again, we find little evidence of lookahead bias.

Lookahead bias may affect stocks to varying degrees depending on the extent of their representation in the pretraining data. Stocks that are frequently mentioned in financial

news, such as large stocks, S&P 500 constituents, or high-media coverage stocks, may be more susceptible to information leakage, potentially giving chronologically inconsistent models an advantage. In particular, earnings announcement days, which typically generate a surge in news coverage, provide a focused setting to test whether inconsistent models gain a disproportionate advantage from information leakage.

In Table D.6, we evaluate long-short portfolio performance across multiple subsamples defined by stock prominence and event-driven attention: S&P 500 membership, large-cap status, high news coverage, and earnings days, each compared against their respective complements. Across all subsamples, we observe comparable performance between ChronoBERT, ChronoGPT, and the chronologically inconsistent models. This suggests no clear evidence of information leakage specific to any of these subsamples.

4 Conclusion

In this paper, we address the challenge of lookahead bias and training leakage in the use of LLMs in social science applications, particularly in financial forecasting. By introducing ChronoBERT and ChronoGPT, two series of chronologically consistent language models trained on timestamped text data, we demonstrate that chronological consistency can be achieved without compromising performance. Our results show that ChronoBERT and ChronoGPT match or surpass the language comprehension abilities of BERT and GPT-2 while generating returns comparable to those of the much larger Llama 3.1 model in asset pricing applications.

Our findings suggest that the impact of lookahead bias in return prediction tasks is modest. We also highlight that the influence of lookahead bias is both model- and application-specific. Notably, downstream predictive models can adapt to limitations in language comprehension, ensuring economically and statistically significant gains.

In addition to quantifying the impact of lookahead bias in return prediction using financial news, we propose a scalable framework for training chronologically consistent LLMs. This framework offers a constructive solution to deal with lookahead bias in predictive modeling, addressing a fundamental challenge in the application of LLMs to finance and other social sciences. By ensuring chronological consistency, our approach lays the foundation for more reliable applications of LLMs in these domains. In subsequent work, we fine-tune ChronoBERT and ChronoGPT to follow instructions with the aim of developing chronologically consistent artificial intelligence (He et al., 2025).

Our work suggests several potential avenues for future research. One avenue would be to assess lookahead bias in other settings. For this purpose, we make our ChronoBERT and ChronoGPT models publicly available at: `https://huggingface.co/manelalab`. Another avenue would be developing compute-optimal training strategies specifically tailored for chronologically consistent LLMs. While we have demonstrated that strong language understanding can be achieved without introducing lookahead bias, further work is needed to establish scaling laws (akin to Chinchilla scaling laws from Hoffmann et al. (2022)) that account for the unique constraints of temporal data limitations (Muennighoff et al., 2023). Such scaling laws would provide guidance on the optimal allocation of computational resources when training models with historical data cutoffs, potentially revealing different optimal ratios of parameters to training tokens compared to models trained on all available data.

References

Bybee, Leland, Bryan Kelly, Asaf Manela, and Dacheng Xiu, 2024, Business news and business cycles, *The Journal of Finance* 79, 3105–3147.

Chen, Yifei, Bryan T. Kelly, and Dacheng Xiu, 2023, Expected returns and large language models, *SSRN Electronic Journal* .

Coleman, Ben, 2020, Why is it okay to average embeddings?, *Randorithms* 16.

Daniel, Kent, David Hirshleifer, and Lin Sun, 2020, Short-and long-horizon behavioral factors, *The Review of Financial Studies* 33, 1673–1736.

Dell, Melissa, Jacob Carlson, Tom Bryan, Emily Silcock, Abhishek Arora, Zejiang Shen, Luca D'Amico-Wong, Quan Le, Pablo Querubin, and Leander Heldring, 2023, American Stories: A Large-Scale Structured Text Dataset of Historical U.S. Newspapers, *Advances in Neural Information Processing Systems* 36, 80744–80772.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019, BERT: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of the 2019 Conference of the Association for Computational Linguistics* 1, 4171–4186.

Didisheim, Antoine, Martina Fraschini, and Luciano Somoza, 2025, Ai's predictable memory in financial analysis, *Economics Letters* 112602.

Drinkall, Felix, Eghbal Rahimikia, Janet B Pierrehumbert, and Stefan Zohren, 2024, Time machine gpt, *arXiv preprint arXiv:2404.18543* .

Dubey, Abhimanyu, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al., 2024, The llama 3 herd of models, *arXiv preprint arXiv:2407.21783* .

Engelberg, Joseph, Asaf Manela, William Mullins, and Luka Vulicevic, 2025, Entity Neutering.

Fama, Eugene F, 1970, Efficient capital markets, *The Journal of Finance* 25, 383–417.

Fama, Eugene F, and Kenneth R French, 2015, A five-factor asset pricing model, *Journal of Financial Economics* 116, 1–22.

Fama, Eugene F, and Kenneth R French, 2018, Choosing factors, *Journal of Financial Economics* 128, 234–252.

Gadre, Samir Yitzhak, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, et al., 2024, Language models scale reliably with over-training and on downstream tasks, *arXiv preprint arXiv:2403.08540* .

Glasserman, Paul, and Caden Lin, 2023, Assessing look-ahead bias in stock return predictions generated by gpt sentiment analysis, *SSRN Electronic Journal* .

Goetzmann, William N, Dasol Kim, and Robert J Shiller, 2022, Crash narratives, Technical report, National Bureau of Economic Research.

Gunasekar, Suriya, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al., 2023, Textbooks are all you need, *arXiv preprint arXiv:2306.11644* .

He, Songrun, Linying Lv, Asaf Manela, and Jimmy Wu, 2025, Chronologically Consistent Generative AI.

He, Songrun, Linying Lv, and Guofu Zhou, 2024, Empirical asset pricing with probability forecasts, *SSRN Electronic Journal* .

Hoberg, Gerard, and Asaf Manela, 2025, The natural language of finance, *Foundations and Trends in Finance* .

Hoffmann, Jordan, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al., 2022, Training compute-optimal large language models, *arXiv preprint arXiv:2203.15556* .

Hou, Kewei, Chen Xue, and Lu Zhang, 2015, Digesting anomalies: An investment approach, *The Review of Financial Studies* 28, 650–705.

Huang, Allen H, Hui Wang, and Yi Yang, 2023, FinBERT: A large language model for extracting information from financial text, *Contemporary Accounting Research* 40, 806–841.

Jha, Manish, Hongyi Liu, and Asaf Manela, 2025, Does finance benefit society? a language embedding approach, *Review of Financial Studies* .

Jiang, Hao, Sophia Zhengzi Li, and Hao Wang, 2021, Pervasive underreaction: Evidence from high-frequency data, *Journal of Financial Economics* 141, 573–599.

Jordan, Keller, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977, 2024, modded-nanogpt: Speedrunning the NanoGPT baseline.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei, 2020, Scaling laws for neural language models, *arXiv preprint arXiv:2001.08361* .

Ke, Zheng Tracy, Bryan T. Kelly, and Dacheng Xiu, 2019, Predicting returns with text data, *SSRN Electronic Journal* .

Kelly, Bryan, Asaf Manela, and Alan Moreira, 2021, Text selection, *Journal of Business & Economic Statistics* 39, 859–879.

Ledoit, Oliver, and Michael Wolf, 2008, Robust performance hypothesis testing with the sharpe ratio, *Journal of Empirical Finance* 15, 850–859.

Levy, Bradford, 2024, Caution Ahead: Numerical Reasoning and Look-ahead Bias in AI Models.

Lo, Kyle, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld, 2020, S2ORC: The semantic scholar open research corpus, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4969–4983.

Lopez-Lira, Alejandro, and Yuehua Tang, 2023, Can ChatGPT forecast stock price movements? return predictability and large language models, *SSRN Electronic Journal* .

Lopez-Lira, Alejandro, Yuehua Tang, and Mingyin Zhu, 2025, The Memorization Problem: Can We Trust LLMs' Economic Forecasts?

Loughran, Tim, and Bill McDonald, 2011, When is a liability not a liability? textual analysis, dictionaries, and 10-ks, *The Journal of Finance* 66, 35–65.

Loureiro, D, F Barbieri, L Neves, L Espinosa-Anke, and J Camacho-Collados, 2022, Timelms: Diachronic language models from twitter. arxiv.

Ludwig, Jens, Sendhil Mullainathan, and Ashesh Rambachan, 2025, Large language models: An applied econometric framework, Technical report, National Bureau of Economic Research.

Lv, Linying, 2024, The value of information from sell-side analysts, *arXiv preprint arXiv:2411.13813* .

Manela, Asaf, and Alan Moreira, 2017, News implied volatility and disaster concerns, *Journal of Financial Economics* 123, 137–162.

Muennighoff, Niklas, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel, 2023, Scaling data-constrained language models, *Advances in Neural Information Processing Systems* 36, 50358–50376.

OLMo, Team, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al., 2024, 2 olmo 2 furious, *arXiv preprint arXiv:2501.00656* .

Penedo, Guilherme, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al., 2024, The fineweb datasets: Decanting the web for the finest text data at scale, *arXiv preprint arXiv:2406.17557* .

Portes, Jacob, Alexander Trott, Sam Havens, Daniel King, Abhinav Venigalla, Moin Nadeem, Nikhil Sardana, Daya Khudia, and Jonathan Frankle, 2023, MosaicBERT: A bidirectional encoder optimized for fast pretraining, *Advances in Neural Information Processing Systems* 36, 3106–3130.

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

Rahimikia, Eghbal, and Felix Drinkall, 2024, Re(visiting) large language models in finance, *SSRN Electronic Journal* .

Sarkar, Suproteem, 2024, StoriesLM: A family of language models with time-indexed training data, *SSRN Electronic Journal* .

Sarkar, Suproteem, and Keyon Vafa, 2024, Lookahead bias in pretrained language models, *SSRN Electronic Journal* .

Silcock, Emily, Abhishek Arora, Luca D'Amico-Wong, and Melissa Dell, 2024, Newswire: A large-scale structured database of a century of historical news, *Advances in Neural Information Processing Systems* 37, 49768–49779.

Tetlock, Paul C., 2007, Giving content to investor sentiment: The role of media in the stock market, *The Journal of Finance* 62, 1139–1168.

Tetlock, Paul C., Maytal Saar-Tsechansky, and Sofus Macskassy, 2008, More than words: Quantifying language to measure firms' fundamentals, *The Journal of Finance* 63, 1437–1467.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman, 2019, GLUE: A multi-task benchmark and analysis platform for natural language understanding, in *Proceedings of the International Conference on Learning Representations* (*ICLR*).

Warner, Benjamin, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al., 2024, Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, *arXiv preprint arXiv:2412.13663* .

Zellers, Rowan, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi, 2019, Hellaswag: Can a machine really finish your sentence?, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* .

Appendix

A Language Understanding Evaluations

A.1 GLUE Evaluation

In this part, we lay out the details of the GLUE (Wang et al., 2019) evaluation process. Following Warner et al. (2024), we use the same evaluation hyperparameters. Here are the details on learning rate, weight decay, and maximum number of epochs for each task. We use early stopping for all the fine-tuning tasks based on validation loss. The RTE, MRPC, and STS-B tasks are fine-tuned starting from the checkpoint of MNLI.

- CoLA (Corpus of Linguistic Acceptability): learning rate: 8e-5; weight decay: 1e-6; maximum epochs: 5.
- SST-2 (Stanford Sentiment Treebank - Binary Classification): learning rate: 8e-5; weight decay: 1e-5; maximum epochs: 2.
- MNLI (Multi-Genre Natural Language Inference): learning rate: 5e-5; weight decay: 5e-6; maximum epochs: 1.
- MRPC (Microsoft Research Paraphrase Corpus): learning rate: 5e-5; weight decay: 5e-6; maximum epochs: 10.
- QNLI (Question Natural Language Inference): learning rate: 8e-5; weight decay: 5e-6; maximum epochs: 2.
- QQP (Quora Question Pairs): learning rate: 5e-5; weight decay: 5e-6; maximum epochs: 10.
- RTE (Recognizing Textual Entailment): learning rate: 5e-5; weight decay: 1e-5; maximum epochs: 3.
- STS-B (Semantic Textual Similarity Benchmark): learning rate: 8e-5; weight decay: 5e-6; maximum epochs: 10.

A.2 Hellaswag Evaluation

To evaluate the ability of our ChronoGPT model to generate coherent and contextually appropriate text, we perform an autoregressive evaluation on the HellaSwag dataset. The goal is to assess how well the model assigns probabilities to different possible sentence completions, with the expectation that the most plausible completion receives the highest probability.

Given a sequence of input tokens $x = (x_1, x_2, \ldots, x_T)$, our model, parameterized by θ, produces a probability distribution over the vocabulary at each time step. The logit output for the sequence is given by:

$$\ell_t = f_\theta(x_1, \ldots, x_{t-1}),$$

where ℓ_t represents the predicted logits for the token x_t.

To compute the autoregressive loss, we shift the token sequence such that the model predicts each token based on previous tokens. The loss for each token is computed using the cross-entropy loss:

$$\mathscr{L}_{b,t} = -\log P_\theta(x_t \mid x_1, \ldots, x_{t-1}),$$

where:

$$P_\theta(x_t \mid x_1, \ldots, x_{t-1}) = \frac{\exp(\ell_{t,x_t})}{\sum_j \exp(\ell_{t,j})}.$$

Since the dataset consists of prompt + completion pairs, we ensure that the evaluation is performed only on the completion tokens. Given a binary mask M of shape (B, T), where $M_{b,t} = 1$ if token $x_{b,t}$ belongs to the completion and 0 otherwise, the masked losses are:

$$\mathscr{L}_{\text{masked},b,t} = M_{b,t} \cdot \mathscr{L}_{b,t}.$$

To compute the total loss per sequence, we sum the loss of each individual completion token in the sequence:

$$L_{\text{sum},b} = \sum_t \mathscr{L}_{\text{masked},b,t}\,.$$

The normalized loss per sequence can be calculated as:

$$L_{\text{avg},b} = \frac{L_{\text{sum},b}}{\sum_t M_{b,t}}\,,$$

where the denominator accounts for the number of valid completion tokens in each sequence.

Since each prompt is associated with multiple completions, we select the completion with the lowest normalized loss as the most probable one:

$$\hat{y} = \arg\min_{b} L_{\text{avg},b}\,.$$

This evaluation framework allows us to assess the model's ability to rank plausible text completions by likelihood. A model with a lower average cross-entropy loss is expected to generate more coherent and contextually appropriate completions.

B Scaling Law of ChronoGPT

A critical question is whether performance can be improved despite the inherent limitations of the chronological training data. According to scaling laws (Kaplan et al., 2020), model performance is a function of parameter count, data size, and compute budget. In our context, the latter two factors face a bottleneck. Scaling the training dataset is challenging due to the strict requirement of chronological consistency. Furthermore, with this constrained dataset, training beyond a certain point (e.g., four epochs) yields modest benefit on loss reduction.

Motivated by the success of model scaling demonstrated by OpenAI (Radford et al., 2019) and AllenAI (OLMo et al., 2024), we investigate if increasing model size is an effective strategy within our data-constrained regime. We test four architectures: 124M, 353M, 1B, and 1.5B parameters, detailed in Table B.1.

Model Size	Vocabulary	Layers	Heads	d_{model}
124 M	50,304	12	6	768
353 M	50,304	24	8	1,024
1 B	50,304	36	12	1,536
1.5 B	50,304	52	12	1,536

Table B.1 Architecture hyperparameters for the four ChronoGPT model sizes.

This table summarizes the core architectural hyperparameters of each scaled-up ChronoGPT model.

We then evaluate them on validation loss and language understanding tasks against our baseline, using a consistent knowledge cutoff of December 1999. As shown in Figure B.1, larger models consistently achieve lower validation loss and higher HellaSwag scores, demonstrating that scaling up model size improves both general language modeling and specific language understanding capabilities. The 1.5B parameter model achieves the best performance across both metrics, with validation loss converging to approximately 1.32 after 80 billion training tokens, compared to 1.35 for the smallest 124M model. Similarly, the 1.5B model reaches a HellaSwag score of 0.37, substantially outperforming the 353M model (0.32) and the smallest 124M model (0.295).

These results confirm that scaling model size is an effective strategy for improving both language modeling and downstream task performance, even within a data-constrained regime. Based on these prominent gains and considering our compute budget, we

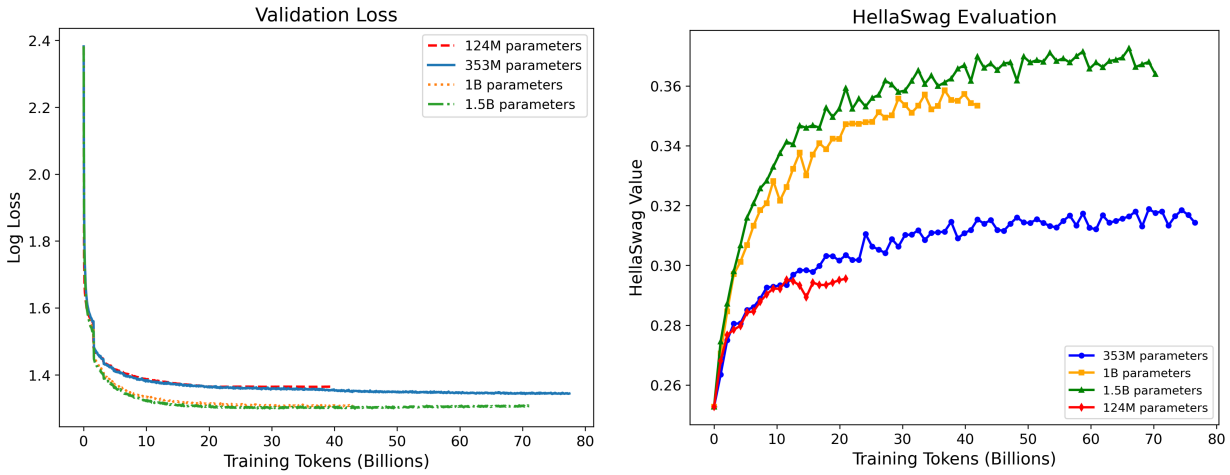


Figure B.1 Validation Loss and Language Understanding Performance Comparison

This figure compares the cross-entropy loss and HellaSwag scores of ChronoGPT models with four sizes: 124 million (124M), 353 million (353M), 1 billion (1B), and 1.5 billion (1.5B) parameters. Validation losses are reported in logs. All models have a consistent knowledge cutoff of December 1999.

select the 1.5B parameter architecture to train the full ChronoGPT series, extending its knowledge base from 1999 to 2024.

Despite the consistent improvement in language performance, we find that asset pricing results remain similar across model sizes. We present the models' performance in Figure 6. Our conclusions about lookahead bias generalize from the 124M model to the 1.5B model, a 12 fold increase in scale.

C Architectural Differences Between BERT and GPT

This section provides a concise overview of the fundamental architectural differences between BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pretrained Transformer), explaining their distinct pretraining objectives and how they handle sequence classification tasks.

C.1 Pretraining Objectives

C.1.1 BERT: Masked Language Modeling (MLM)

BERT employs a bidirectional approach where it randomly masks some percentage of input tokens (30% for ChronoBERT) and then predicts those masked tokens. For a given input sequence $X = (x_1, x_2, ..., x_n)$, some tokens are replaced with a special [MASK] token, creating a partially masked sequence X_{masked}. The pretraining objective is to predict the original tokens at the masked positions:

$$\mathscr{L}_{MLM} = \mathbb{E}_{(X,m)\sim\mathscr{D}} \left[-\sum_{i \in m} \log P(x_i | X_{masked}) \right],$$

where m is the set of masked token indices and $\mathscr{D}$ is the training distribution.

C.1.2 GPT: Causal Language Modeling (CLM)

GPT uses an autoregressive approach for pretraining, predicting the next token given all previous tokens in the sequence. For an input sequence $X = (x_1, x_2, ..., x_n)$, the model minimizes:

$$\mathscr{L}_{CLM} = -\sum_{i=1}^{n} \log P(x_i | x_1, x_2, ..., x_{i-1}).$$

This objective trains the model to generate coherent text by predicting each token based only on its preceding context.

C.2 Attention Mechanisms

The key architectural difference between BERT and GPT lies in their attention mechanisms:

C.2.1 BERT: Bidirectional Self-Attention

In BERT, each token attends to all tokens in the sequence, regardless of position. The self-attention computation for token i at layer ℓ is:

$$\text{Attention}(Q_i^\ell, K^\ell, V^\ell) = \text{softmax}\left(\frac{Q_i^\ell \cdot (K^\ell)^T}{\sqrt{d_k}}\right) \cdot V^\ell,$$

where Q_i^ℓ is the query vector for token i, and K^ℓ and V^ℓ are the key and value matrices for all tokens in the sequence. This allows BERT to incorporate context from both before and after each token, capturing bidirectional relationships.

C.2.2 GPT: Causal Self-Attention

GPT uses a causal (or masked) self-attention mechanism where each token can only attend to itself and previous tokens:

$$\text{CausalAttention}(Q_i^\ell, K^\ell, V^\ell) = \text{softmax}\left(\frac{Q_i^\ell \cdot (K_{1:i}^\ell)^T}{\sqrt{d_k}}\right) \cdot V_{1:i}^\ell.$$

This is achieved by masking future positions in the attention matrix, effectively limiting the receptive field to tokens $j \leq i$.

C.3 Sequence Classification Approaches

C.3.1 BERT for Sequence Classification

BERT performs sequence classification by utilizing the first token of the input sequence. After passing through the BERT transformer layers, the final hidden state of this token serves as a comprehensive representation of the entire sequence:

$$h_{[CLS]} = \text{BERT}(x_1, x_2, ..., x_n)_1.$$

A classification head (typically a linear layer followed by softmax) is then applied:

$$P(y|X) = \text{softmax}(W \cdot h_{[CLS]} + b).$$

During fine-tuning, the entire model, including the pretrained transformer and the

classification head, is updated to minimize the cross-entropy loss:

$$\mathscr{L}_{cls} = -\sum_{c=1}^{C} y_c \log(P(y_c|X)),$$

where C is the number of classes and y_c is the ground truth label.

C.3.2 GPT for Sequence Classification

GPT handles sequence classification differently from BERT. Instead of using the first token, GPT leverages the last token as a representation of the entire sequence, as it attends to all preceding tokens. After passing through the GPT transformer layers, the final hidden state of the last token serves as a comprehensive sequence representation:

$$h_{[CLS]} = \text{GPT}(x_1, x_2, ..., x_n)_n.$$

Once the sequence representation is extracted, the classification head and loss functions remain the same as in BERT.

These architectural differences explain the performance variations observed between ChronoBERT and ChronoGPT in our experiments. While both models achieve chronological consistency, their underlying architectures make them suited to different types of tasks, with ChronoBERT excelling at classification and understanding benchmarks (GLUE) and ChronoGPT being better adapted to generative and completion tasks (HellaSwag).

C.4 Pretraining Hyperparameters

This section provides the full set of pretraining hyperparameters for ChronoBERT and ChronoGPT. ChronoBERT is based on the ModernBERT architecture (Warner et al., 2024), and ChronoGPT builds on modded-nanoGPT (Jordan et al., 2024). Key deviations from each reference implementation are noted at the bottom of each table.

Table C.1 provides the details for hyperparameters for ChronoBERT and Table C.2 provides the details for hyperparameters for ChronoGPT.

Hyperparameter	Value
Base implementation	ModernBERT
Tokenizer	ModernBERT-base
Vocabulary size	50,368
Context length	1,024 tokens
Layers / Heads / d_{model}	22 / 12 / 768
FFN intermediate size	1,152
Global batch size	4,718,592 tokens
Optimizer	Decoupled StableAdamW
Peak learning rate	8×10^{-4}
(β_1, β_2)	(0.9, 0.98)
Weight decay	1×10^{-5}
LR schedule	Linear warmup $\rightarrow$ constant $\rightarrow$ linear warmdown
Warmup duration	3B tokens
Masking rate	30% (train) / 15% (eval)
Attention	Sliding window (128) + global every 3 layers
Precision	amp_bf16
Base model training tokens	460B (7B unique $\times$ 66 epochs)
Stopping rule	Loss / GLUE plateau
Deviations from ModernBERT	
Context length	1,024 (vs. 8,192)
Incremental training	Year-by-year checkpoint chaining (2000–2024)
Incremental LR schedule	No warmup or decay (constant LR)
LR	8×10^{-4}

Table C.1 ChronoBERT Pretraining Hyperparameters

This table reports the pretraining hyperparameters for ChronoBERT. The base model (ChronoBERT_{1999}) is trained on the pre-2000 corpus. Incremental vintage models (2000–2024) inherit the previous year's checkpoint and train on that year's new data with the deviations noted above.

Hyperparameter	Value
Base implementation	modded-nanoGPT
Tokenizer	GPT-2
Vocabulary size	50,304
Context length	1,792 tokens
Layers / Heads / d_{model}	52 / 12 / 1,536
FFN Intermediate Size	6,144
Global batch size	524,288 tokens
Optimizer (2D matrix params)	Muon (momentum = 0.95, Nesterov)
Optimizer (embeddings, LM head)	AdamW (fused)
Peak LR — Muon	0.05
Peak LR — embeddings	0.6
Peak LR — LM head	0.008
(β_1, β_2) for AdamW	(0.9, 0.95)
LR schedule	Linear warmup $\rightarrow$ constant $\rightarrow$ linear warmdown
Warmup duration	0.27B tokens
Weight decay	0.01
Precision	bfloat16
Base model training tokens	70B (7B unique $\times$ 10 epochs)
Stopping rule	Loss / HellaSwag plateau
Deviations from modded-nanoGPT	
Model scale	1.5B parameters (vs. 124M)
Incremental training	Year-by-year checkpoint chaining (2000–2024)
Incremental LR schedule	No warmup or decay (constant LR)
LR — Muon	0.025
LR — embeddings	0.3
LR — LM head	0.004

Table C.2 ChronoGPT Pretraining Hyperparameters

This table reports the pretraining hyperparameters for ChronoGPT (1.5B). The base model (ChronoGPT_{1999}) is trained on the pre-2000 corpus. Incremental vintage models (2000–2024) inherit the previous year's checkpoint and train on that year's new data. The Muon optimizer applies orthogonal momentum updates (via Newton–Schulz iteration) to 2D weight matrices, while AdamW handles embeddings, scalar parameters, and the language modeling head.

D Additional Results

D.1 Testing Lookahead Bias using Chrono$_{t-1}$ vs. Chrono$_t$ Models

	ChronoBERT$_{t-1}$			ChronoBERT$_t$			ChronoGPT$_{t-1}$			ChronoGPT$_t$		
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
Low (L)	-23.30	25.86	-0.90	-21.30	25.91	-0.82	-19.82	25.15	-0.79	-19.28	25.19	-0.77
2	-2.43	25.20	-0.10	-4.55	25.30	-0.18	1.48	25.48	0.06	-0.62	25.65	-0.02
3	4.17	25.64	0.16	3.21	25.41	0.13	0.61	24.87	0.02	0.39	25.07	0.02
4	4.17	24.58	0.17	2.28	24.59	0.09	2.41	24.26	0.10	0.89	24.25	0.04
5	3.94	24.22	0.16	7.91	24.23	0.33	7.18	24.54	0.29	7.11	24.62	0.29
6	10.81	24.13	0.45	11.33	24.15	0.47	5.78	24.20	0.24	6.62	24.10	0.27
7	14.56	24.23	0.60	12.43	24.15	0.51	10.82	23.84	0.45	10.88	23.77	0.46
8	16.38	23.64	0.69	16.68	23.68	0.70	15.74	24.30	0.65	17.84	24.25	0.74
9	23.95	24.45	0.98	23.79	24.35	0.98	22.63	24.19	0.94	23.54	23.93	0.98
High (H)	37.71	24.53	1.54	38.23	24.50	1.56	43.08	25.41	1.70	42.57	25.61	1.66
H-L	61.02	12.72	4.80	59.52	12.83	4.64	62.90	12.78	4.92	61.85	13.09	4.72

Table D.1 Performance of the LLM Portfolios

This table presents annualized performance metrics (mean return, standard deviation, and Sharpe ratio) for decile portfolios sorted by next-day return predictions from financial news. Portfolios are rebalanced daily, with the "H-L" row representing a strategy of longing the top decile and shorting the bottom decile. All values are in percentage points except Sharpe ratios. All portfolios are equal-weighted. Data spans January 2008–July 2023.

	ChronoBERT$_{t-1}$	ChronoBERT$_t$	ChronoGPT$_{t-1}$	ChronoGPT$_t$
ChronoBERT$_{t-1}$	–	0.054	0.717	0.379
ChronoBERT$_t$	0.946	–	0.888	0.641
ChronoGPT$_{t-1}$	0.283	0.112	–	0.053
ChronoGPT$_t$	0.621	0.359	0.947	–

Table D.2 P-value of Pairwise SR Difference Tests

This table reports the p-value from the Ledoit and Wolf (2008) Sharpe ratio difference test of the 'H-L' portfolios from different LLMs in Table D.1. Each entry corresponds to a test of the null hypothesis that the Sharpe ratio of the model in the row is smaller than or equal to that of the model in the column. The portfolio sample spans from January 2008 to July 2023.

D.2 Embedding Strategy Selection

Model	First Layer	Average Layers	Last Layer
$\text{ChronoBERT}_{\text{Realtime}}$	0.973	0.000	0.027
$\text{ChronoGPT}_{\text{Realtime}}$	0.678	0.272	0.050
Llama 3.1	0.979	0.021	0.000
GPT-2 XL	0.021	0.979	0.000
ModernBERT	0.995	0.000	0.005
BERT	0.048	0.947	0.005
FinBERT	0.037	0.000	0.963
StoriesLM	0.920	0.032	0.048

Table D.3 Fraction of Months Each Embedding Strategy is Selected

This table reports the fraction of months in which each embedding extraction strategy is selected for each language model. Three strategies are considered: using hidden states from the first layer, averaging hidden states across all layers, and using hidden states from the last layer. The sample spans January 2008 to July 2023.

D.3 Predicting Stock Returns: Tercile Portfolios

	$\text{ChronoBERT}_{\text{Realtime}}$			$\text{ChronoGPT}_{\text{Realtime}}$			Llama 3.1			GPT-2 XL		
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
Low (L)	-6.22	24.46	-0.25	-4.70	24.16	-0.19	-8.22	24.48	-0.34	-6.61	24.34	-0.27
2	8.51	23.65	0.36	5.98	23.52	0.25	9.46	23.68	0.40	8.43	23.62	0.36
High (H)	24.98	23.50	1.06	25.98	23.89	1.09	26.02	23.50	1.11	25.44	23.67	1.07
H-L	31.20	7.26	4.30	30.68	6.86	4.47	34.24	7.36	4.65	32.05	7.21	4.45
	ModernBERT			BERT			FinBERT			StoriesLM		
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
Low (L)	-6.53	24.52	-0.27	-6.82	24.43	-0.28	-5.88	24.75	-0.24	-4.77	24.40	-0.20
2	10.03	23.72	0.42	11.16	23.59	0.47	10.52	23.55	0.45	8.78	23.53	0.37
High (H)	23.77	23.38	1.02	22.94	23.60	0.97	22.62	23.35	0.97	23.26	23.71	0.98
H-L	30.30	7.24	4.19	29.76	7.40	4.02	28.50	7.72	3.69	28.03	7.56	3.71

Table D.4 Performance of the LLM Portfolios: Tercile Portfolios

This table presents annualized performance metrics (mean return, standard deviation, and Sharpe ratio) for decile portfolios sorted by next-day return predictions from financial news. Portfolios are rebalanced daily, with the "H-L" row representing a strategy of longing the top decile and shorting the bottom decile. All values are in percentage points except Sharpe ratios. All portfolios are equal-weighted. Data spans January 2008–July 2023.

D.4 Predicting Stock Returns: Full News Text

	ChronoBERT$_{\text{Realtime}}$			ChronoGPT$_{\text{Realtime}}$			Llama 3.1			GPT-2 XL		
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
Low (L)	-24.93	26.36	-0.95	-23.72	27.20	-0.87	-33.10	27.80	-1.19	-29.59	27.46	-1.08
2	-1.30	24.88	-0.05	-2.84	25.39	-0.11	-3.01	26.15	-0.12	-3.85	25.11	-0.15
3	3.86	25.23	0.15	1.44	24.33	0.06	2.08	24.65	0.08	0.73	24.45	0.03
4	9.54	24.58	0.39	5.61	24.09	0.23	8.93	24.39	0.37	7.49	24.42	0.31
5	9.92	24.10	0.41	5.81	24.02	0.24	12.50	24.05	0.52	11.84	24.52	0.48
6	11.62	24.16	0.48	13.35	24.42	0.55	13.15	23.64	0.56	16.39	24.27	0.68
7	12.73	23.51	0.54	14.86	24.01	0.62	17.17	23.89	0.72	15.25	23.87	0.64
8	16.28	23.96	0.68	19.46	24.00	0.81	19.02	23.71	0.80	19.13	23.64	0.81
9	24.39	24.06	1.01	25.68	24.15	1.06	24.21	23.81	1.02	24.67	23.67	1.04
High (H)	28.93	25.26	1.15	31.40	24.60	1.28	30.11	24.63	1.22	28.99	24.79	1.17
H-L	53.86	13.40	4.02	55.12	14.30	3.85	63.20	14.97	4.22	58.58	14.02	4.18
	ModernBERT			BERT			FinBERT			StoriesLM		
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
Low (L)	-20.33	26.62	-0.76	-22.54	27.14	-0.83	-23.67	27.10	-0.87	-19.88	26.77	-0.74
2	-3.84	25.54	-0.15	-0.33	24.93	-0.01	0.55	25.87	0.02	0.74	25.32	0.03
3	1.74	25.04	0.07	1.50	24.40	0.06	3.89	24.92	0.16	3.65	24.58	0.15
4	8.15	24.25	0.34	4.07	24.34	0.17	7.21	24.11	0.30	7.38	24.42	0.30
5	10.82	24.35	0.44	12.17	24.29	0.50	10.77	24.01	0.45	7.91	23.79	0.33
6	12.71	24.29	0.52	10.75	23.67	0.45	12.20	23.82	0.51	13.97	24.12	0.58
7	15.69	23.92	0.66	15.76	23.92	0.66	16.78	24.18	0.69	13.24	23.83	0.56
8	15.81	23.92	0.66	19.66	24.03	0.82	17.18	23.79	0.72	19.28	24.05	0.80
9	21.57	23.84	0.90	22.18	24.22	0.92	20.53	23.87	0.86	21.26	24.44	0.87
High (H)	28.69	24.43	1.17	27.81	25.25	1.10	25.63	24.25	1.06	23.54	24.87	0.95
H-L	49.03	13.98	3.51	50.35	14.05	3.58	49.30	13.82	3.57	43.42	13.46	3.23

Table D.5 Performance of the LLM Portfolios: Full News Text

This table presents annualized performance metrics (mean return, standard deviation, and Sharpe ratio) for decile portfolios sorted by next-day return predictions from the full text of financial news. To ensure the most important information is captured within each language model's context window, embeddings are formed by placing all headlines ahead of all the main text for each stock-day observation. Portfolios are rebalanced daily, with the "H-L" row representing a strategy of longing the top decile and shorting the bottom decile. All values are in percentage points except Sharpe ratios. All portfolios are equal-weighted. Data spans January 2008–July 2023.

D.5 Predicting Stock Returns: Subsamples

	S&P 500			non-S&P 500		
	mean	SD	SR	mean	SD	SR
ChronoBERT	7.49	11.30	0.66	75.03	15.03	4.99
ChronoGPT	7.10	11.33	0.63	76.18	15.28	4.99
Llama 3.1	7.24	12.34	0.59	80.87	15.80	5.12
GPT-2 XL	7.83	11.01	0.71	73.62	15.49	4.75
ModernBERT	5.92	11.54	0.51	72.69	15.40	4.72
	Large market cap			Small market cap		
	Mean	SD	SR	Mean	SD	SR
ChronoBERT	9.76	9.30	1.05	98.11	20.87	4.70
ChronoGPT	11.82	9.51	1.24	96.97	20.64	4.70
Llama 3.1	12.48	10.30	1.21	104.12	21.05	4.95
GPT-2 XL	9.54	9.52	1.00	96.36	21.37	4.51
ModernBERT	9.42	9.59	0.98	92.44	20.97	4.41
	High news coverage			Low news coverage		
	Mean	SD	SR	Mean	SD	SR
ChronoBERT	30.67	13.68	2.24	81.65	17.94	4.55
ChronoGPT	34.65	14.20	2.44	79.45	18.00	4.41
Llama 3.1	33.23	14.81	2.24	88.26	18.68	4.72
GPT-2 XL	32.14	14.55	2.21	82.75	18.63	4.44
ModernBERT	29.73	14.04	2.12	79.15	18.30	4.33
	Earnings Days			Non-Earnings Days		
	Mean	SD	SR	Mean	SD	SR
ChronoBERT	67.86	40.36	1.68	55.18	13.37	4.13
ChronoGPT	72.18	39.14	1.84	57.25	13.27	4.31
Llama 3.1	51.90	51.26	1.01	62.99	15.20	4.14
GPT-2 XL	55.88	37.04	1.51	55.28	14.12	3.92
ModernBERT	58.84	42.25	1.39	55.58	13.89	4.00

Table D.6 Performance of the LLM Portfolios: Subsamples

This table reports the performance of the "High-minus-Low" (H–L) portfolio across various subsamples. The subsamples are based on:

- Index Membership: S&P 500 constituents vs. non-S&P 500 stocks.
- Market Capitalization: Firms are split into large-cap and small-cap groups, using the median market capitalization at the end of the previous month as the cutoff.
- Recent News Coverage: Stocks are divided into high and low news coverage groups, based on the median number of news articles published in the previous month.
- Earnings Day News vs. Non-Earnings Day News: Returns are further split by whether the news occurred within or beyond a three-day window around earnings announcement days ([t-1, t+1]).

D.6 News Incorporation

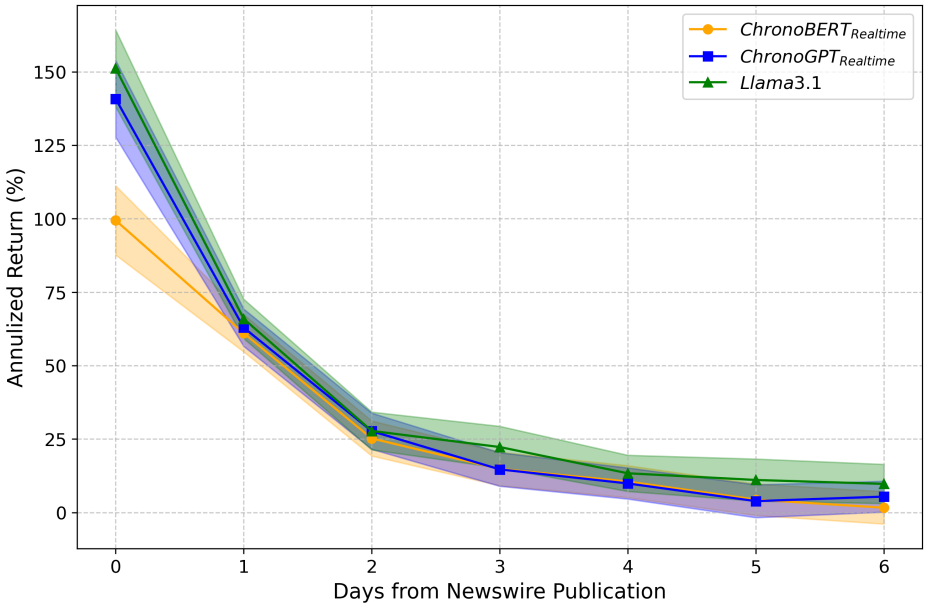


Figure D.1 Speed of News Incorporation

This figure compares the annualized returns (%) of ChronoBERT, ChronoGPT, and Llama 3.1 portfolios over a 7-day period. Day 0 marks the first trading day contemporaneous with the news announcement. The performance shown for these LLM portfolios in the analyses above specifically reflects their return on Day 1. Each point reflects the equal-weighted average return of a long-short decile portfolio formed by sorting stocks based on the respective language model's next-day return predictions. Shaded areas represent 95% confidence intervals.