# Chronologically Consistent Large Language Models*

Songrun He Linying Lv Asaf Manela Jimmy Wu

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

**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, we find that ChronoBERT's real-time outputs achieve a Sharpe ratio comparable to state-of-the-art models, indicating that lookahead bias 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

*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 trained 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 state-of-the-art 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), even our earliest ChronoBERT outperforms existing no-leakage models, including StoriesLM (Sarkar and Vafa, 2024) and FinBERT (Huang et al., 2023). It also matches or surpasses the popular BERT (Devlin et al., 2019), which ranks first in downloads among all language models on Hugging Face as of February 2025.[1] Strong language understanding, we show, does not require data from the future.

On the HellaSwag sentence-completion task (Zellers et al., 2019), all ChronoGPT vintages surpass the language understanding ability of OpenAI's GPT of similar size. Thus 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.

Among all social sciences, finance is particularly sensitive to lookahead bias. Market efficiency tests (Fama, 1970) assume that prices reflect only known facts 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 by forecasting stock returns from news with ChronoBERT and ChronoGPT. Leveraging extensive financial newswire data, we find that the portfolio performance of ChronoBERT and ChronoGPT matches that of the state-of-the-art Llama 3.1 (Dubey et al., 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 context is relatively modest.

An important observation is that the impact of lookahead bias is model- and application-specific. While ChronoBERT and ChronoGPT may exhibit lower language comprehension

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

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.

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 open-weight baselines. Third, we measure lookahead bias in news-based return forecasting 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), 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), which can destroy information. Instead, we train a series of chronologically consistent models with knowledge cutoffs from 1999 to 2024 which offer superior language understanding and more up-to-date knowledge to similar such 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 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 state-of-the-art BERT architecture from Portes et al. (2023) and Warner et al. (2024).[3] Compared to the original BERT model by

[2]In contemporaneous work, Rahimikia and Drinkall (2024) introduce a series of 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 ChronoBERT, which achieves an SR of 4.50 over the same period. Unfortunately, the authors have since retracted their models from Hugging Face, preventing further direct comparison.

[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.

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 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.

## 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

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

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

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

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] 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 and contextual structures. Our first model checkpoint, $\text{ChronoGPT}_{1999}$, is trained on 21 billion tokens over approximately 3 epochs, 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

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

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.

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. Because HellaSwag employs adversarial filtering to create highly deceptive distractors, strong performance on this task signals robust contextual and procedural reasoning.[9]

[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 B describes the architectural differences between ChronoBERT and ChronoGPT for sequence classification in GLUE evaluation.

[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.

### 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 economic 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 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 of 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:

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

[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.

[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 $\lambda$ chosen from grid points ranging from $10^{-10}$ to $10^{10}$.

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.

## 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

ChronoBERT$_{1999}$ and ChronoGPT$_{1999}$ are pretrained for multiple epochs on a corpus of 7 billion tokens comprising chronologically organized English text sourced from diverse domains, including historical web content, archived news articles, and scientific publications. The dataset is fully open-source and is carefully curated to include only high-quality text published before the year 2000, ensuring a focus on no leakage of future knowledge. The final composition of our pretraining corpus was determined through extensive ablation studies to optimize model performance.

We further conduct incremental training from 2000 to 2024 on a corpus of 65 billion tokens with similar high-quality, diverse, timestamped, and open-source textual data to update knowledge for the model.

### 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 Wall Street Journal, Barron's, and MarketWatch.

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 four other language models. We then validate the chronological consistency of our models. Finally, we assess its 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 1, validation loss (measured via cross-entropy) decreases consistently as the number of training tokens increases.[13] Simultaneously, language prediction accuracy improves with training progress.

These improvements translate into enhanced language understanding, as reflected in the GLUE and HellaSwag scores. ChronoBERT begins outperforming BERT after approximately 350 billion training tokens and continues to improve thereafter, while ChronoGPT surpasses a GPT-2 model of the same size after 21 billion training tokens.[14]

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 2 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 2, the GLUE and HellaSwag scores for nearly all models exceed that of BERT and GPT-2, highlighting their superior language understanding and overall quality.

[13]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.

[14]We obtained the Hellaswag evaluation score for the GPT-2 model from Karpathy (2024), available at https://github.com/karpathy/build-nanogpt/blob/master/play.ipynb.

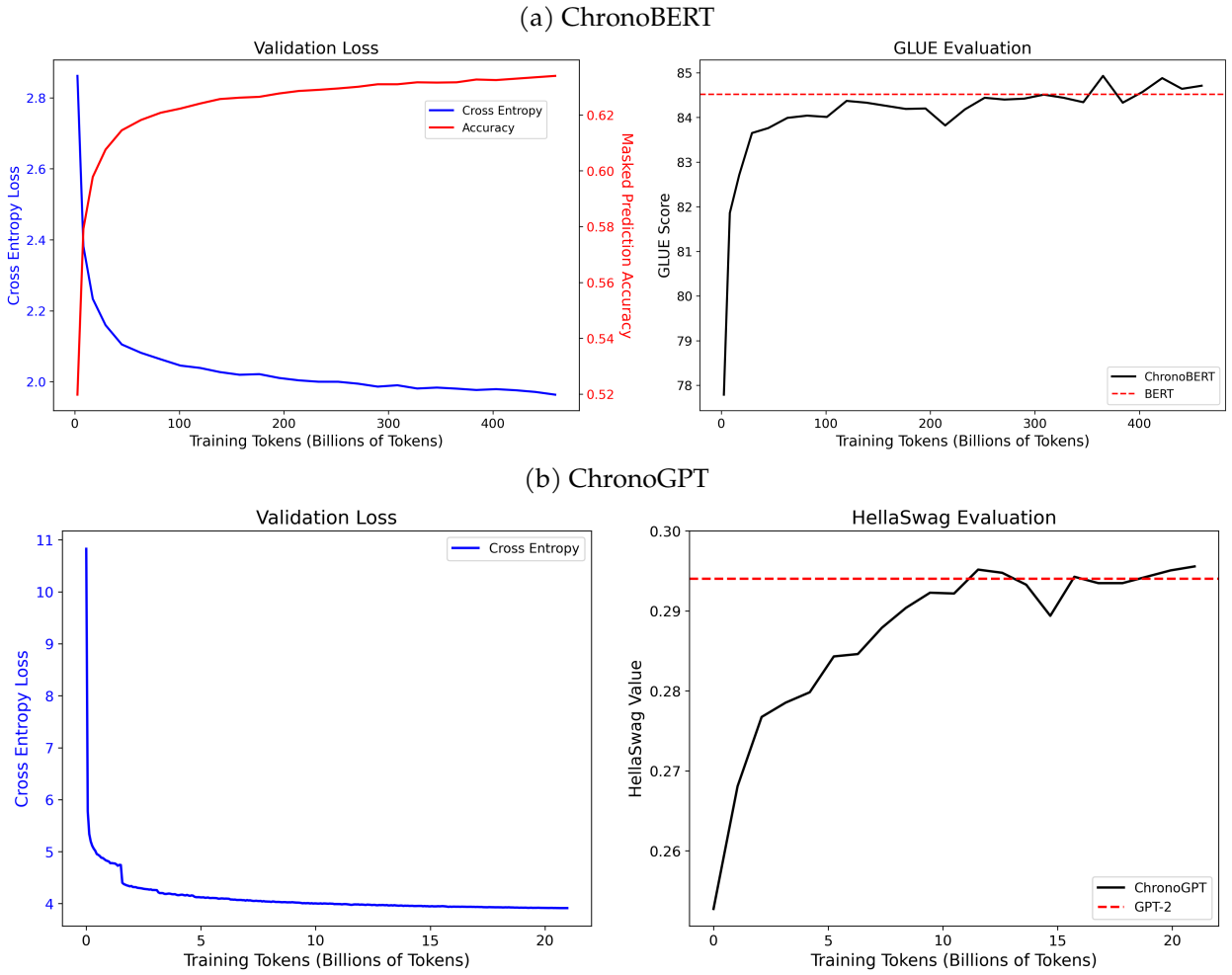



**Figure 1** Validation Loss and Evaluation Scores versus Pretraining Tokens

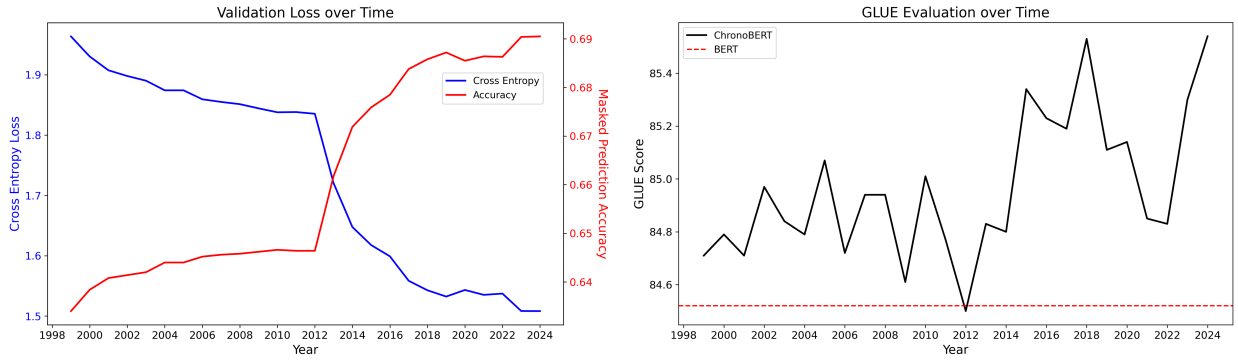
The left panel shows the validation loss, measured using cross-entropy loss and language prediction accuracy, as the ChronoBERT$_{1999}$ and ChronoGPT$_{1999}$ are trained on an increasing number of tokens. The right panel displays the GLUE scores and HellaSwag values as training progresses. For ChronoBERT and ChronoGPT, the final model checkpoint is trained on 460 billion tokens and 21 billion tokens, respectively. The training corpus consists of text up to December 1999.

## 3.2 Language Understanding

We further compare language understanding against other models. Table 1 summarizes key characteristics of these models, including parameter counts, context lengths, and knowledge cutoffs.

The models evaluated include:

(a) ChronoBERT




(b) ChronoGPT

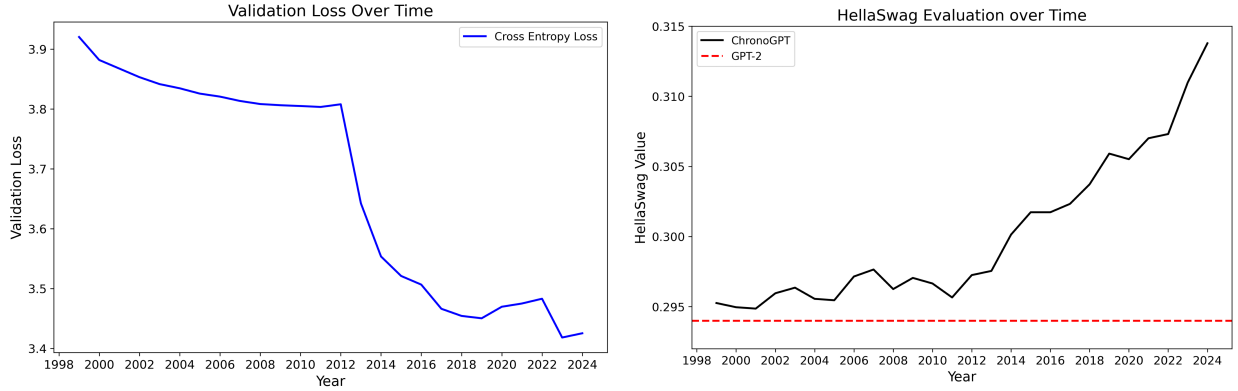



**Figure 2** Validation Loss and Evaluation Scores over Time

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.

**ChronoBERT**$_t$: Our initial BERT-based model, ChronoBERT$_{1999}$ is pretrained on 460 billion tokens of pre-2000, diverse, high-quality, and open-source text data 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 data available in year $t$. Our final checkpoint of the ChronoBERT series, ChronoBERT$_{2024}$, is pretrained on 525 billion tokens of diverse, high-quality, and open-source text until December 2024.

**ChronoGPT**$_t$: Our initial GPT-based model, trained with the same chronologically consistent text data as ChronoBERT, ensures no lookahead bias by only utilizing information available at each point in time. The ChronoGPT series spans from 1999 to 2024,

with each yearly checkpoint building upon the previous year's model.

**BERT**: The original BERT model trained on Wikipedia and the BookCorpus dataset by Devlin et al. (2019).[15]

**FinBERT**: A domain-specific model pretrained on financial texts, including regulatory filings, analyst reports, and earnings call transcripts, from Huang et al. (2023).[16]

**StoriesLM**: A pretrained model from Sarkar and Vafa (2024), trained on historical news articles. We use the final version trained on data up to 1963.[17]

**GPT-2**: The 124 million parameter version of OpenAI's GPT-2 model from Radford et al. (2019).[18]

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

| | Parameters | Context Tokens | Knowledge Cutoff |
|---|---|---|---|
| $\text{ChronoBERT}_{1999}$ | 149M | 1,024 | December, 1999 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| $\text{ChronoBERT}_{2024}$ | 149M | 1,024 | December, 2024 |
| $\text{ChronoGPT}_{1999}$ | 124M | 1,792 | December, 1999 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| $\text{ChronoGPT}_{2024}$ | 124M | 1,792 | December, 2024 |
| BERT | 110M | 512 | October, 2018 |
| FinBERT | 110M | 512 | December, 2019 |
| StoriesLM | 110M | 512 | December, 1963 |
| GPT-2 | 124M | 1,024 | Febrary, 2019 |
| Llama 3.1 | 8,030M | 128,000 | December 2023 |

**Table 1** 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.

[15]Model downloaded from https://huggingface.co/google-bert/bert-base-uncased.
[16]Model downloaded from https://huggingface.co/yiyanghkust/finbert-pretrain.
[17]Model downloaded from https://huggingface.co/StoriesLM/StoriesLM-v1-1963.
[18]Model downloaded from https://huggingface.co/openai-community/gpt2.
[19]Model downloaded from https://huggingface.co/meta-llama/Llama-3.1-8B.

| | $\text{ChronoBERT}_{1999}$ | $\text{ChronoBERT}_{2024}$ | $\text{ChronoGPT}_{1999}$ | $\text{ChronoGPT}_{2024}$ |
|---|---|---|---|---|
| COLA | 57.32 | 56.32 | 37.13 | 31.70 |
| SST2 | 91.82 | 92.58 | 89.68 | 88.53 |
| MRPC | 92.71 | 92.45 | 82.92 | 85.34 |
| STSB | 89.57 | 89.93 | 81.57 | 82.58 |
| QQP | 88.54 | 88.90 | 82.43 | 83.53 |
| MNLI | 86.19 | 86.89 | 77.63 | 79.15 |
| QNLI | 90.61 | 92.04 | 84.94 | 85.98 |
| RTE | 80.94 | 85.20 | 67.08 | 67.80 |
| GLUE | 84.71 | 85.54 | 75.42 | 75.58 |
| | Llama 3.1 | BERT | FinBERT | StoriesLM |
| COLA | 55.86 | 57.59 | 28.99 | 46.85 |
| SST2 | 95.49 | 92.62 | 89.03 | 90.44 |
| MRPC | 88.22 | 90.76 | 88.59 | 89.33 |
| STSB | 90.67 | 90.07 | 85.72 | 87.01 |
| QQP | 89.67 | 88.21 | 86.60 | 86.88 |
| MNLI | 89.59 | 84.98 | 79.23 | 79.78 |
| QNLI | 95.35 | 91.52 | 86.12 | 87.44 |
| RTE | 85.63 | 80.43 | 67.00 | 67.15 |
| GLUE | 86.31 | 84.52 | 76.41 | 79.36 |

**Table 2** 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.

Table 2 displays the GLUE scores for eight different models. While Llama 3.1 achieves the highest overall GLUE score (86.31), $\text{ChronoBERT}_{2024}$ and $\text{ChronoBERT}_{1999}$, 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 (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.

By contrast, ChronoGPT$_{2024}$ and ChronoGPT$_{1999}$ do not match ChronoBERT's scores on GLUE. Their lower performance reflects the nature of an autoregressive (decoder-only) architecture, which processes text strictly left to right for text generation. While this approach can excel at generative tasks, it often underperforms on classification benchmarks that benefit from bidirectional processing of the entire input.

Overall, ChronoBERT exhibits strong language understanding, 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'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 also match or exceed the performance of BERT and GPT-2 despite their chronologically bounded pretraining data, demonstrating that high-quality LLMs can be constructed without chronological leakage. 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 it a robust choice for a wide range of downstream tasks.

## 3.3 Validation of Chronological Consistency

To detect leakage in the textual data used to pretrain our chronologically consistent models, we evaluate them on events occurring after the model's knowledge cutoff. For ChronoBERT, this involves constructing a sentence with a masked token representing

| Prompt year: | 1992 | 2000 | 2008 | 2016 | 2020 | 2024 |
|---|---|---|---|---|---|---|
| BERT | **Clinton** | Clinton | **Obama** | Obama | Obama | Obama |
| ChronoBERT$_{1999}$ | Roosevelt | Roosevelt | Roosevelt | Hoover | Hoover | Washington |
| ChronoBERT$_{2000}$ | **Clinton** | Clinton | Clinton | Clinton | Clinton | Wilson |
| ChronoBERT$_{2001}$ | **Clinton** | Clinton | Clinton | Clinton | Clinton | Washington |
| ChronoBERT$_{2002}$ | **Clinton** | **Bush** | Bush | Clinton | Bush | Clinton |
| ChronoBERT$_{2003}$ | **Clinton** | **Bush** | Bush | Clinton | Bush | Clinton |
| ChronoBERT$_{2004}$ | **Clinton** | **Bush** | Bush | Clinton | Bush | Clinton |
| ChronoBERT$_{2005}$ | **Clinton** | **Bush** | Bush | Bush | Bush | Clinton |
| ChronoBERT$_{2006}$ | **Clinton** | **Bush** | Bush | Bush | Bush | Clinton |
| ChronoBERT$_{2007}$ | **Clinton** | **Bush** | Bush | Bush | Bush | Monroe |
| ChronoBERT$_{2008}$ | **Clinton** | **Bush** | Bush | Obama | Bush | Wilson |
| ChronoBERT$_{2009}$ | **Clinton** | Clinton | **Obama** | Obama | Obama | Wilson |
| ChronoBERT$_{2010}$ | **Clinton** | Obama | **Obama** | Obama | Obama | Wilson |
| ChronoBERT$_{2011}$ | **Clinton** | Clinton | **Obama** | Obama | Obama | Wilson |
| ChronoBERT$_{2012}$ | Obama | Obama | **Obama** | Obama | Obama | Obama |
| ChronoBERT$_{2013}$ | **Clinton** | Obama | **Obama** | Obama | Obama | Monroe |
| ChronoBERT$_{2014}$ | **Clinton** | **Bush** | **Obama** | Obama | Obama | Monroe |
| ChronoBERT$_{2015}$ | **Clinton** | Clinton | **Obama** | Obama | Obama | Monroe |
| ChronoBERT$_{2016}$ | **Clinton** | **Bush** | **Obama** | Obama | Obama | Obama |
| ChronoBERT$_{2017}$ | **Clinton** | **Bush** | **Obama** | **Trump** | Trump | Monroe |
| ChronoBERT$_{2018}$ | **Clinton** | **Bush** | **Obama** | **Trump** | Trump | Obama |
| ChronoBERT$_{2019}$ | **Clinton** | **Bush** | **Obama** | **Trump** | Trump | Obama |
| ChronoBERT$_{2020}$ | **Clinton** | **Bush** | **Obama** | **Trump** | Trump | **Trump** |
| ChronoBERT$_{2021}$ | **Clinton** | Clinton | **Obama** | **Trump** | **Biden** | Biden |
| ChronoBERT$_{2022}$ | **Clinton** | **Bush** | **Obama** | **Trump** | **Biden** | Biden |
| ChronoBERT$_{2023}$ | **Clinton** | **Bush** | **Obama** | **Trump** | **Biden** | Biden |
| ChronoBERT$_{2024}$ | **Clinton** | **Bush** | **Obama** | **Trump** | **Biden** | Biden |

**Table 3** Mask Language Modeling-Based Predictions of U.S. Presidents

This table displays ChronoBERT's masked-token predictions to the sentence **"After the {year} U.S. presidential election, President [MASK] was inaugurated as U.S. President in the year {year+1}."** The gray 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. For reference, predictions from BERT (released in October 2018) are also included.

time-specific information beyond the model's cutoff. We use each model to tokenize the sentence and predict the masked token. For example, consider:

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

If a model is well-trained with high-quality and time-relevant textual data, it should be able to identify any president elected 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.

Table 3 presents the model predictions, with the gray area in the top-right indicating predictions in the post-knowledge cutoff period and the non-shaded lower-right denoting the predictions strictly in the pre-knowledge cutoff period. Correct predictions are highlighted in blue. In the pre-cutoff period, the ChronoBERT models correctly identify a majority of presidents, outperforming the original BERT model—which was correct only twice out of four attempts in its knowledge window. In contrast, during the post-cutoff period, none of the ChronoBERT models correctly predicts a future president in his first term. The sole correct prediction is for $\text{ChronoBERT}_{2020}$, which suggests President Trump's second non-consecutive term after the 2024 election; this reflects a tendency to favor either a recently elected or historically notable president in an out-of-knowledge context. Overall, these findings validate that the textual data used to train our chronologically consistent models contains no evidence of leakage.

## 3.4 Predicting Stock Returns using Financial News

To quantify the economic gains from enhanced language understanding, we analyze stock return predictions 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.[20]

Table 4 presents the decile portfolio performance for realtime models ($\text{ChronoBERT}_{\text{Realtime}}$

[20]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.

| | ChronoBERT$_{\text{Realtime}}$ | | | ChronoGPT$_{\text{Realtime}}$ | | | Llama 3.1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Mean | SD | SR | Mean | SD | SR | Mean | SD | SR |
| Low(L) | -23.30 | 25.86 | -0.90 | -20.03 | 25.96 | -0.77 | -23.71 | 26.15 | -0.91 |
| 2 | -2.43 | 25.20 | -0.10 | 0.06 | 25.65 | 0.00 | -4.77 | 25.31 | -0.19 |
| 3 | 4.17 | 25.64 | 0.16 | 2.96 | 25.03 | 0.12 | -0.24 | 24.86 | -0.01 |
| 4 | 4.17 | 24.58 | 0.17 | 5.59 | 24.75 | 0.23 | 3.84 | 24.62 | 0.16 |
| 5 | 3.94 | 24.22 | 0.16 | 6.67 | 24.36 | 0.27 | 7.47 | 24.65 | 0.30 |
| 6 | 10.81 | 24.13 | 0.45 | 5.91 | 23.91 | 0.25 | 12.03 | 24.23 | 0.50 |
| 7 | 14.56 | 24.23 | 0.60 | 13.51 | 24.09 | 0.56 | 13.31 | 24.33 | 0.55 |
| 8 | 16.38 | 23.64 | 0.69 | 16.63 | 23.77 | 0.70 | 15.13 | 23.79 | 0.64 |
| 9 | 23.95 | 24.45 | 0.98 | 21.56 | 24.07 | 0.90 | 24.68 | 23.88 | 1.03 |
| High(H) | 37.71 | 24.53 | 1.54 | 37.13 | 24.59 | 1.51 | 42.20 | 25.05 | 1.68 |
| H-L | 61.02 | 12.72 | 4.80 | 57.16 | 12.75 | 4.48 | 65.91 | 13.46 | 4.90 |
| | BERT | | | FinBERT | | | StoriesLM | | |
| | Mean | SD | SR | Mean | SD | SR | Mean | SD | SR |
| Low(L) | -22.52 | 26.21 | -0.86 | -23.96 | 26.86 | -0.89 | -17.80 | 26.52 | -0.67 |
| 2 | -5.05 | 25.55 | -0.20 | -3.17 | 25.64 | -0.12 | -1.19 | 25.26 | -0.05 |
| 3 | 3.12 | 24.92 | 0.13 | 3.36 | 24.83 | 0.14 | 1.86 | 24.92 | 0.07 |
| 4 | 8.14 | 24.62 | 0.33 | 7.19 | 24.52 | 0.29 | 5.90 | 24.62 | 0.24 |
| 5 | 10.81 | 24.44 | 0.44 | 9.17 | 24.39 | 0.38 | 4.99 | 24.30 | 0.21 |
| 6 | 9.38 | 24.02 | 0.39 | 11.47 | 24.03 | 0.48 | 11.88 | 23.90 | 0.50 |
| 7 | 14.54 | 23.83 | 0.61 | 16.54 | 23.92 | 0.69 | 12.41 | 23.66 | 0.52 |
| 8 | 18.51 | 24.04 | 0.77 | 19.16 | 23.65 | 0.81 | 18.93 | 24.19 | 0.78 |
| 9 | 19.68 | 23.90 | 0.82 | 20.70 | 23.88 | 0.87 | 23.25 | 24.30 | 0.96 |
| High(H) | 33.37 | 24.88 | 1.34 | 29.51 | 24.60 | 1.20 | 29.73 | 24.78 | 1.20 |
| H-L | 55.89 | 13.38 | 4.18 | 53.47 | 13.85 | 3.86 | 47.53 | 13.90 | 3.42 |

**Table 4** 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.

and ChronoGPT$_{\text{Realtime}}$) For example, in the year $t$, we would use the latest available 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 four other benchmarks: (1) BERT; (2) FinBERT; (3) StoriesLM; and (4) Llama-3.1-8B. 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 4.48, outperforming StoriesLM, FinBERT and BERT. These results demonstrate that increased language understanding indeed translates into significant economic gains.

| | ChronoBERT | ChronoGPT | Llama 3.1 | BERT | FinBERT | StoriesLM |
|---|---|---|---|---|---|---|
| ChronoBERT | | 0.076 | 0.685 | 0.005 | 0.002 | 0.000 |
| ChronoGPT | 0.924 | | 0.973 | 0.078 | 0.017 | 0.001 |
| Llama 3.1 | 0.315 | 0.027 | | 0.001 | 0.000 | 0.000 |
| BERT | 0.995 | 0.922 | 0.999 | | 0.116 | 0.005 |
| FinBERT | 0.998 | 0.983 | 1.000 | 0.884 | | 0.098 |
| StoriesLM | 1.000 | 0.999 | 1.000 | 0.995 | 0.902 | |

**Table 5** 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 4. Each entry corresponds to a test of the null hypothesis that the Sharpe ratio of the model in the row is smaller than 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 5. Comparing realtime ChronoBERT and ChronoGPT against StoriesLM, we find that the models' improved language understanding and superior knowledge increases 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 state-of-the-art Llama 3.1 model, we find that ChronoBERT generates a comparable Sharpe ratio, with no statistically significant differences (p-value of 0.315), while ChronoGPT significantly underperforms Llama (p-value of 0.027). Between the two chronological models themselves, we observe a modest statistical difference (p-value of 0.076), indicating that both approaches effectively leverage temporal information.

These findings suggest that the performance of our chronologically consistent ChronoBERT model is comparable to state-of-the-art LLMs in this financial application, despite its limited training using historical textual data. While ChronoGPT may be a better choice for text generation tasks, when it is used for embeddings which are fed into a prediction model, as we do here, ChronoBERT appears to be a slightly better choice.

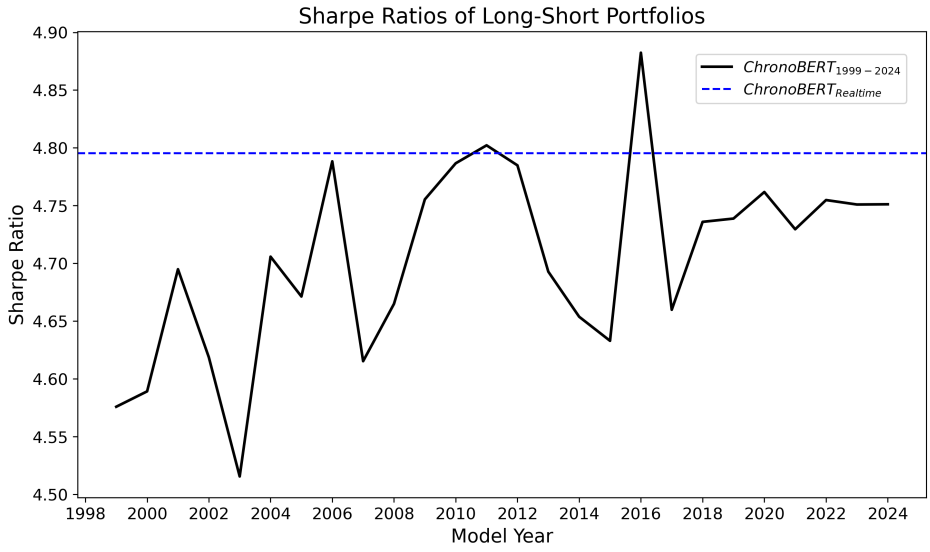
A somewhat surprising finding is that lookahead bias appears to be modest in this stock return forecasting application. The fact that the Llama-based performance is no different from that of the chronologically consistent ChronoBERT, suggests that understanding the news flow into the market in realtime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

Figure 3 further presents the trading performance of the whole series of chronologically consistent models. Specifically, for each time $t$ in the figure, we use the time $t$ model to embed news articles and run predictions using embeddings from the model. We find consistent performance across all models in the series. The results again highlight (1) the return prediction exercise has modest lookahead bias; (2) the enhanced language understanding shown in Figure 2 indeed translates into significant economic gains.

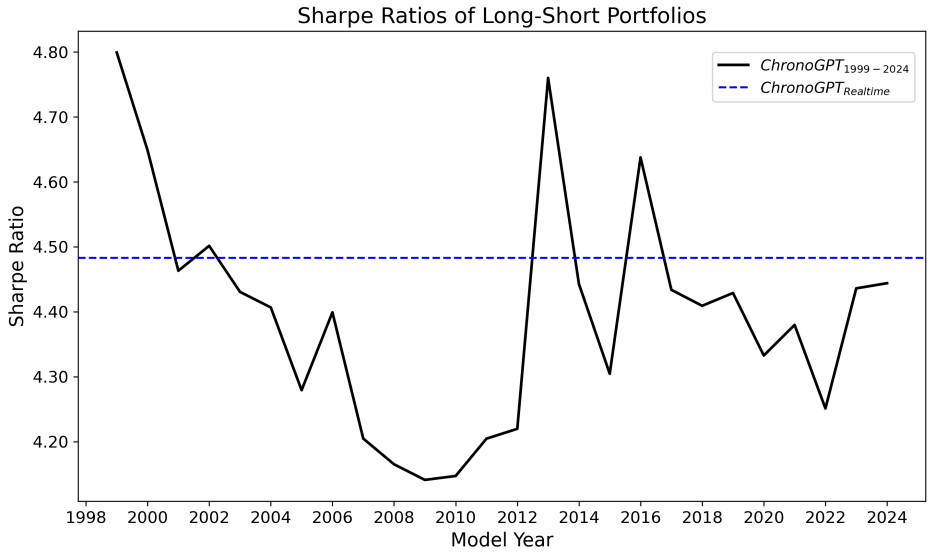
# 4 Conclusion

In this paper, we address the critical challenge of lookahead and survivorship bias in LLMs used 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 performance comparable to the much larger Llama 3.1 model in asset pricing applications.

Our findings reveal 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.




(a) ChronoBERT




(b) ChronoGPT

**Figure 3** Portfolios Performance across ChronoBERT and ChronoGPT Vintages

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 realtime models.

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.

We make our ChronoBERT and ChronoGPT models publicly available at: https://huggingface.co/manelalab.

Our work suggests several potential avenues for future research. One direction 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.

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.

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* .

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

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

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, 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* .

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.

Karpathy, Andrej, 2024, Build NanoGPT.

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.

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

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.

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.

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* .

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 finetuned 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 $\theta$, 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 $\ell_t$ 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 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 Pre-trained Transformer), explaining their distinct pretraining objectives and how they handle sequence classification tasks.

## B.1 Pretraining Objectives

### B.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.

### B.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.

## B.2 Attention Mechanisms

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

### B.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 $\ell$ 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 $Q_i^\ell$ is the query vector for token $i$, and $K^\ell$ and $V^\ell$ 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.

### B.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 $j \leq i$.

## B.3 Sequence Classification Approaches

### B.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.

### B.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).