Chronologically Consistent Generative AI *

Songrun He Linying Lv Asaf Manela Jimmy Wu

First draft: July 2025.

Abstract

We introduce a family of chronologically consistent, instruction-following large language models to eliminate lookahead bias. Each model is trained only on data available before a clearly defined knowledge-cutoff date, ensuring strict temporal separation from any post-cutoff data. The resulting framework offers (i) a simple, conversational chat interface, (ii) fully open, fixed model weights that guarantee replicability, and (iii) a conservative lower bound on forecast accuracy, isolating the share of predictability that survives once training leakage is removed. Together, these features provide researchers with an easy-to-use generative AI tool useful for a wide range of prediction tasks that is free of lookahead bias.

JEL Classification: G11, G12, G17

Keywords: Instruction following model, chronological consistency, lookahead bias, training leakage

*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).

1 Introduction

Large language models (LLMs) have emerged as a transformative force in financial econometrics. Recent research extensively leverages LLM outputs for prediction and estimation tasks (see, for example: Lopez-Lira and Tang, 2023; Chang et al., 2023; Jha et al., 2024; Chen et al., 2025; Lv, 2025). The novel capabilities and inherent intelligence of LLMs have enabled exploration of unstructured data and addressing previously unanswered questions.

However, as noted by Sarkar and Vafa (2024) and Ludwig et al. (2025), most prediction problems with generative AI face methodological challenges of lookahead bias. The reason is that LLMs are pretrained on a vast corpus of text data that incorporates future information relative to the prediction task. This leads to lookahead bias when the model's knowledge cutoff τ extends beyond the prediction time, as a training leakage term then emerges in the loss function.

Several papers have introduced robust methods for isolating temporal information. For example, Glasserman and Lin (2023) and Engelberg et al. (2025) develop a systematic entity masking approach so that LLMs cannot recognize firms. Alternatively, researchers have explored pretraining LLMs from scratch, meticulously curating the training data to control the models' inherent knowledge (e.g., Sarkar, 2024; He et al., 2025).

While pretraining an LLM from scratch appears to be a more comprehensive solution for preventing the model from accessing future knowledge, as opposed to methods that attempt to make it forget learned information or infer from context, its implementation presents two primary challenges. First, ensuring chronological consistency requires excluding a substantial amount of future training text, which may compromise the model's performance. Second, integrating this approach, especially within a model embedding and machine learning pipeline, is both technically sophisticated and computationally expensive.

In this paper, we offer another solution more accessible to the social science research community: the first instruction-following chat model free of lookahead bias. Specifically, during both the pretraining and instruction fine-tuning stages, we carefully curate the dataset to prevent the model from seeing future knowledge. For example, ChronoGPT-Instruct$_\tau$ (where τ is in $\{1999, 2005, 2010, 2015, 2020\}$) never accesses any knowledge that emerged or became economically salient after τ. For any evaluation set post-τ, the model has perfect temporal separation from the evaluation set.

Despite the significant effort in developing the ChronoGPT-Instruct series, certain challenges in its design and implementation are important to highlight. A fundamental tradeoff exists between maintaining robust chat capabilities and ensuring strict chronological consistency. To illustrate, a Qwen-1.5-1.8B-Chat model of similar parameter size is pretrained on 2.2 trillion tokens, approximately 314 times the 7 billion tokens seen by our base model. Nevertheless, most ChronoGPT-Instruct models achieve above 10% win rates in the Alpaca instruction-following evaluation, demonstrating their practical utility despite data constraints. Additionally, while our prompt-based filtering algorithm is designed to be highly effective, we acknowledge it is not theoretically flawless. However, in a rigorous validation test we conduct, ChronoGPT-Instruct models consistently fail to predict future presidents or major events, showing no signs of training leakage.

The primary contribution of our model is to serve as a robust tool for conducting lookahead bias-free robustness tests in various prediction problems. While ChronoGPT-Instruct does not offer a perfect solution that simultaneously eliminates training leakage and preserves state-of-the-art language abilities, it allows for establishing a conservative lower bound of predictive power, providing a clearer understanding of true model performance. In a prompt-based trading portfolio example we provide, if ChronoGPT-Instruct is considered a lookahead bias-free counterpart to larger models such as Qwen-1.5-1.8B-Chat and Llama-3.2-3B-Instruct (up to twice the parameter count and trained on far more data), our finding implies that at least 73% of the observed news return

predictability persists without leakage. The remaining discrepancy in Sharpe ratios (e.g., between 1.19 and 1.62) likely stems from a combination of differences in model capabilities and lookahead bias in the comparison model.

2 Methodology and Data

In this section, we describe our instruction finetuning methodology designed to enforce no training leakage, followed by details of our data curation process and the datasets used for both instruction finetuning and return prediction tasks.

2.1 Instruction Finetuning

Our instruction finetuning pipeline is designed to satisfy the *no–training-leakage* contract of Ludwig et al. (2025). The contract formalizes the idea that any text used for evaluating a model must be statistically independent of the text used to train it. We first restate the contract in a two-stage setting of pretraining and instruction finetuning (IFT), and then show how our data curation enforces each of its requirements.

Two disjoint training corpora. Let

$$t^{\text{pre}}(\tau) = \{\sigma \in \Sigma^* : \text{date}(\sigma) \leq \tau\}, \quad t^{\text{ift}}(\tau) = \{\sigma \in \Sigma^*_{\text{inst}} : \text{date}(\sigma) \leq \tau\}, \tag{1}$$

where τ is the knowledge cutoff of our vintage models. For any text piece r define stage-specific indicators

$$t_r^{\text{pre}} = \mathbf{1}\big(r \in t^{\text{pre}}(\tau)\big), \quad t_r^{\text{ift}} = \mathbf{1}\big(r \in t^{\text{ift}}(\tau)\big) \tag{2}$$

and let

$$t_r = \max\left\{t_r^{\text{pre}}, t_r^{\text{ift}}\right\} \in \{0,1\} \tag{3}$$

denote membership in the combined training set.

Loss decomposition with a leakage term. Consider an evaluation sample $R_{>\tau}$ consisting exclusively of documents dated after τ and let $D_r = \mathbf{1}(r \in R_{>\tau})$. With loss function $\ell(\cdot,\cdot)$ and model prediction $\hat{m}(r;t)$, the expectation of $\hat{L}_\tau$ can be written as

$$E\left[\hat{L}_\tau\right] = \underbrace{E\left[\ell\left(Y_r, \hat{m}(r;t)\right)\right]}_{\text{true out-of-sample loss}} \underbrace{- E\left[D_r\left(\frac{q_{T|D}\left(t_r\right)}{q_T\left(t_r\right)} - 1\right)\ell\left(Y_r, \hat{m}(r;t)\right)\right]}_{\text{leakage term}}, \tag{4}$$

where

$$q_T\left(t_r\right) = \Pr\left(t_r = 1\right), \quad q_{T|D}\left(t_r\right) = \Pr\left(t_r = 1 \mid D_r = 1\right) \tag{5}$$

are the *unconditional* and *conditional* probabilities that r appears in the training set, respectively. The second expectation in (4) is the leakage term; it vanishes if and only if

$$\forall r : \frac{q_{T|D}\left(t_r\right)}{q_T\left(t_r\right)} = 1, \tag{6}$$

which is the contract's independence condition.

Stage-wise sufficiency. Because the overall indicator t_r in (3) is the union of two disjoint events, independence is guaranteed once it holds separately for pre-training and IFT:

$$\forall r : \frac{q_{T|D}\left(t_r^{\text{pre}}\right)}{q_T\left(t_r^{\text{pre}}\right)} = 1 \quad \text{and} \quad \frac{q_{T|D}\left(t_r^{\text{ift}}\right)}{q_T\left(t_r^{\text{ift}}\right)} = 1. \tag{7}$$

For the pretraining stage, we use the vintage ChronoGPT in He et al. (2025) as our base model. The corpus T_τ^{pre} is built from historical web snapshots, archived news, and scientific literature. Every document carries a verifiable publication timestamp, and any text dated after τ is discarded. Hence for each post-knowledge cutoff evaluation item r we have $t_r^{\text{pre}} = 0$, leading to

$$q_T\left(t_r^{\text{pre}}\right) = 0, \quad q_{T|D}\left(t_r^{\text{pre}}\right) = 0, \tag{8}$$

so the first equality in (7) holds.

Prompt: You are provided with a user-assistant interaction. Your task is to determine whether the conversation contains any information that would have been unavailable or irrelevant prior to the year 2000.

Specifically, indicate whether the message includes any direct or indirect reference to:

1. A concept, company, product, technology, event, online review, or terminology that was created, discovered, or publicly introduced after 1999, or

2. A subject that only gained significant economic, cultural, scientific, or technological relevance after 1999, even if it existed before that date.

If such a reference is present anywhere in the conversation, return: 1

If the conversation is entirely composed of content that could have been generated using only knowledge available prior to 2000, return: 0

Clarifications:

- For conversations evaluated as low quality, also assign a label of 1.

- In cases of uncertainty or ambiguity, adopt a conservative approach and assign a label of 1.

- References to post-1999 entities such as GPT models, Kubernetes, TikTok, blockchain, COVID-19, Tesla, or similar modern constructs are strong indicators of a label of 1.

Return your answer strictly as a JSON object with the following fields:

- ”label”: either 0 or 1

- ”confidence”: a number from 0 to 10 (higher means more certain)

- ”suspected term”: a brief phrase (1-3 words) that triggered your label decision, or ”none” if label=0

Example output: {”label”: 1, ”confidence”: 9, ”suspected term”: ”GPT-3”}

Here is the message:{conversation}

We conduct a prompt filtering algorithm for the instruction-finetuning stage. Candidate instruction-response pairs are screened with a large-language-model classifier. The classifier, implemented with ChatGPT-4.1, receives the following prompt and returns a binary label indicating whether the dialogue contains any knowledge that emerged or

became economically salient after τ.

Only pairs receiving the label 0 are admitted to $t^{\text{ift}}(\tau)$. Consequently, for every evaluation item r dated after the cut-off, $t_r^{\text{ift}} = 0$ and the second equality in (7) also holds. We conduct a validation test for the independence condition in section 3.2.

2.2 Data

This section introduces the public user-assistant interaction dataset we use for instruction finetuning, and the financial newswire data we use for return prediction.

2.2.1 Instruction Finetuning Data

The instruction-fine-tuning corpus comprises over 100,000 prompt–response pairs drawn from three public resources and arranged as a curriculum that grows in cognitive load and sequence length. We start with simple, short tasks from the LLMs-from-Scratch book, like spelling checks or basic math. Then, we add medium-length prompts from the GPT-3-generated Self-Instruct dataset. Finally, we include the broad AllenAI's Tulu-3 SFT mixture.

All entries are filtered to (i) exclude non-English records and code snippets, and (ii) satisfy a temporal-knowledge screen: each example is classified by GPT-4.1 and only retained when the model assigns label 0 ("knowledge available pre-2000") with the maximum confidence score of 10. Table 1 summarizes the resulting dataset.

Stage	SFT data	Number of examples	Average conversation length
1	LLMs-from-scratch	1,097	102
2	GPT-3 self-generated	67,136	183
3	Tulu-3 SFT mixture	32,293	2,513

Table 1 Instruction Finetuning Datasets

We then format these entries as inputs to ChronoGPT using Alpaca-style prompt formatting. Below is an example entry passed to the LLM:

```
{
  Below is an instruction that describes a task. Write a response that
  appropriately completes the request.
    ### Instruction:
    Identify the correct spelling of the following word.
    ### Input:
    Ocassion
    ### Response:
    The correct spelling is 'Occasion.'
}
```

2.2.2 Financial Newswire Data

We use the Dow Jones Newswire dataset, a real-time newswire providing extensive coverage of financial markets from January 2007 to July 2023. This dataset includes news headlines, full article texts, and precise display timestamps. Following He et al. (2025), we focus on firm-specific news, aggregating all relevant headlines for each firm within a trading day. Finally, we merge this news data with CRSP close-to-close returns on day t+1 to examine the predictability of stock returns.

3 Results

3.1 Instruction Following Evaluation

We instruction-finetune a series of models as ChronoGPT-Instruct$_{1999}$, ChronoGPT-Instruct$_{2005}$, ChronoGPT-Instruct$_{2010}$, ChronoGPT-Instruct$_{2015}$, ChronoGPT-Instruct$_{2020}$, each starting from the corresponding ChronoGPT vintage model in He et al. (2025). All vintages are fine-tuned with the standard masked cross-entropy for next-token prediction, formally defined as

$$\mathscr{L} = -\frac{1}{N}\sum_{t=1}^{N} \log p_\theta\left(y_t \mid \mathbf{x}_{<t}\right), \tag{9}$$

where the model parameters θ are optimized to maximize the probability $p_\theta\left(y_t \mid \mathbf{x}_{<t}\right)$ of generating the true token y_t at position t within a sequence of length N. This objective rewards the model for accurately predicting the subsequent token in a response.

Every SFT stage logs the token-level cross-entropy on a 5% hold-out split that is never seen by the optimizer. Figure 1 plots these losses across three stages of training, showing the characteristic rapid decline in early steps followed by more gradual improvement. The steep initial drop reflects rapid adaptation to the instruction-following format, while later stages show continued learning from the curriculum order.

Figure 2 compares the validation losses across all five vintages for each training stage. Consistent with the language-model results in He et al. (2025), we observe systematic improvements from earlier to later vintages, with the 1999 model showing the highest validation loss and more recent vintages achieving lower losses across all three stages.

While validation loss is a useful proxy, it conflates syntax prediction with pragmatic instruction-following. We therefore evaluate using a head-to-head comparison against Qwen-1.5-1.8B-Chat on Alpaca length-controlled (LC) evaluation. For each instruction in the AlpacaEval dataset, we generated outputs from both our model and the reference

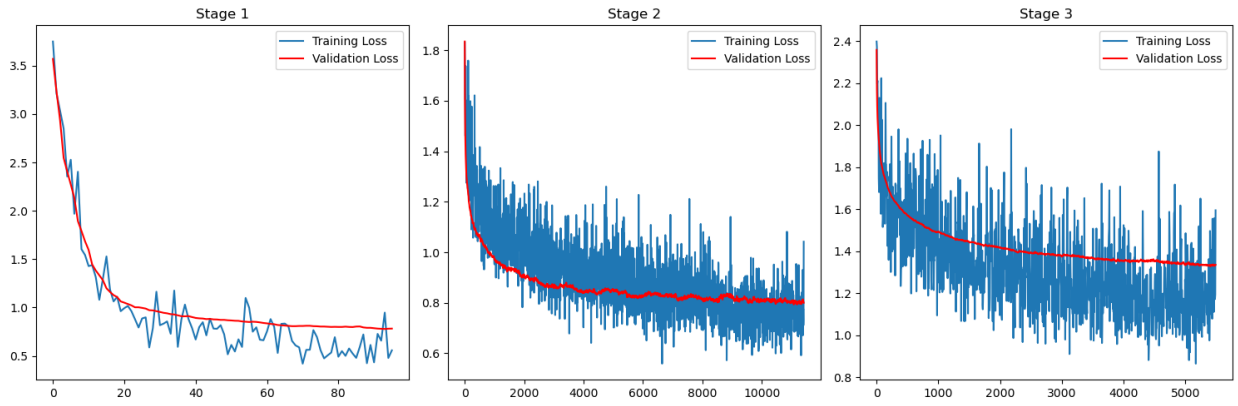


Figure 1 Training Loss and Validation Loss of Instruction Finetuning

The figure shows the training dynamics across three stages of supervised fine-tuning (SFT) on the ChronoGPT-1999 model. Stage 1 uses LLMs-from-scratch data. Stage 2 uses GPT-3 self-generated data. Stage 3 sees Tulu-3-SFT mixture.

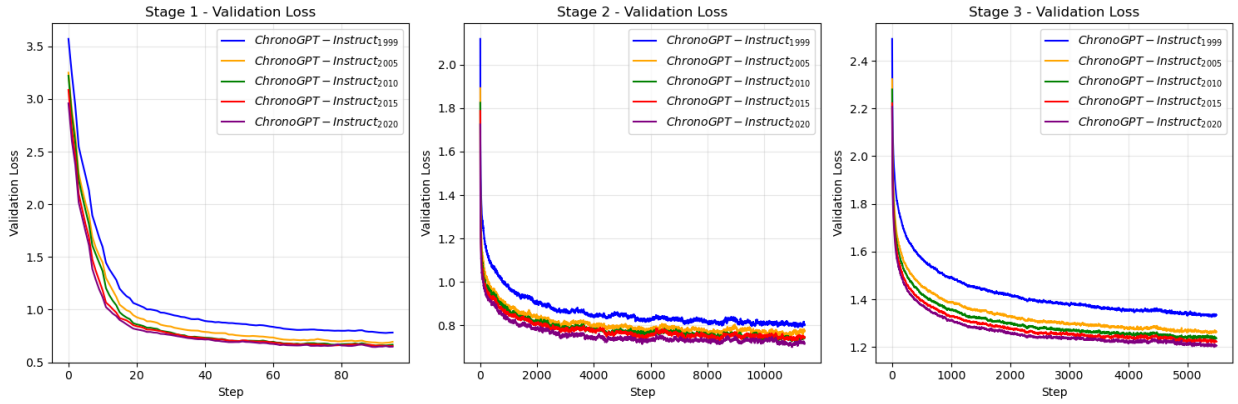


Figure 2 Validation Loss of Instruction Model Vintages

The figure compares the validation loss across five vintage models (1999, 2005, 2010, 2015, 2020) for the three curriculum stages: Stage 1 (LLMs-from-scratch, simple tasks), Stage 2 (GPT-3 self-generated, medium complexity), and Stage 3 (Tulu-3 mixture, complex conversations).

model. These pairs of outputs were then presented to an automatic evaluator, which determined a preference. A 50% win rate indicates that the evaluated model performs comparably to the reference model.

Figure 3 reports the resulting win rates across vintages. ChronoGPT-Instruct$_{1999}$ achieves a 6.73% win rate. Performance increases monotonically to 10.23% in 2005,

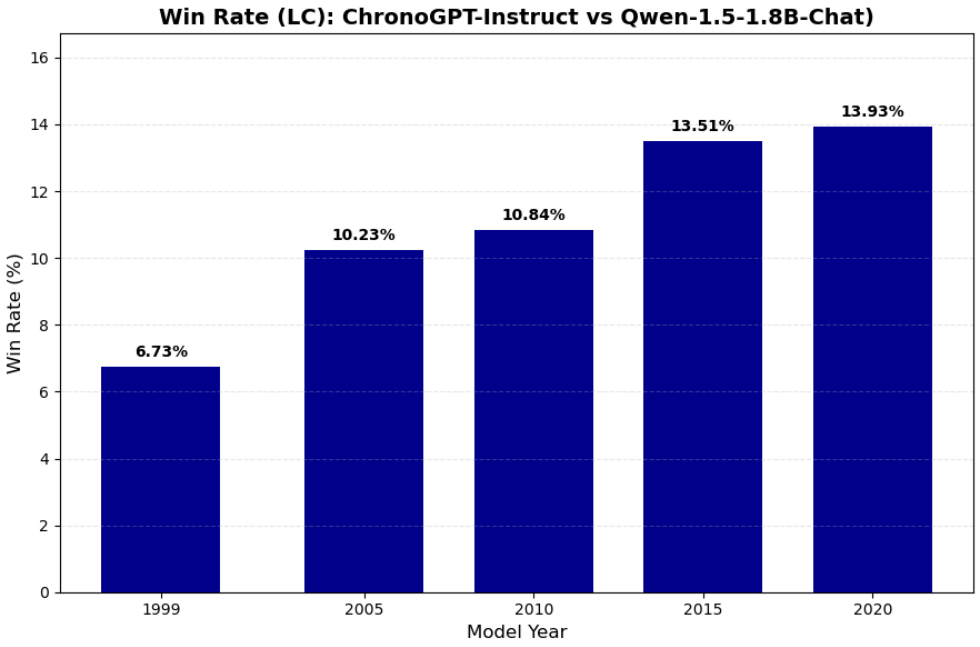


Figure 3 Alpaca Evaluation for ChronoGPT-Instruct

This figure shows head-to-head win rates using length-controlled evaluation across model vintages from 1999 to 2020. The evaluation set is from AlpacaFarm. The benchmark model is Qwen-1.5-1.8B-Chat.

10.84% in 2010, 13.51% in 2015, and peaks at 13.93% in 2020. This steady improvement demonstrates that more recent training data consistently enhances instruction-following capabilities, with the 2020 vintage showing the strongest performance. Despite these gains, the overall low win rates largely stem from the significant disparity in pretraining data volume: the reference Qwen-1.5-1.8B-Chat model was pretrained on approximately 314 times more tokens than our base ChronoGPT models.

3.2 Chronological Consistency Validation

While we take great care in curating our training data to ensure it includes only information believed to be available as of a specific date, the process is not immune to error. Such errors may originate in either the pre-training or fine-tuning datasets. In the case of

pre-training data, inaccuracies in recorded publication dates—such as those introduced when printed materials are digitized via optical character recognition (OCR) and assigned incorrect timestamps—can lead to the inadvertent inclusion of information that was not actually available at the intended time. For fine-tuning, classification was performed using GPT-4.1, which may misidentify whether certain prompt-response pairs reflect knowledge from before or after the year 2000. These issues can result in lookahead bias, unintentionally exposing the model to information that should not have been available at the specified training cutoff.

For the pre-training dataset, He et al. (2025) test for leakage in the ChronoGPT series using textual sequences involving U.S. presidents and major events from various years, and find no evidence of leakage. In this section, we replicate their validation exercise for ChronoGPT-Instruct to test whether there is additional leakage introduced in the instruction fine-tuning process.

	Prompt year						Accuracy	
	1993	2001	2009	2017	2021	2025	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
Llama-3.2-3B-Instruct	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	**Joe Biden**	R.	5/5	0/1
Qwen-1.5-1.8B-Chat	**Bill Clinton**	**George W**	**Barack Obama**	**Donald Trump**	**Joe Biden**	Kamala	5/5	0/1
ChronoGPT-Instruct$_{1999}$	**Bill Clinton**	Bill Clinton	Clinton\n	Obama Obama	Obama Obama	John F.	1/1	0/5
ChronoGPT-Instruct$_{2005}$	**Bill Clinton**	**George W.**	George W.	George W.	George W.	George W.	2/2	0/4
ChronoGPT-Instruct$_{2010}$	**Bill Clinton**	**George W.**	**Barack Obama**	Barack Obama	Barack Obama	George W.	3/3	0/3
ChronoGPT-Instruct$_{2015}$	**Bill Clinton**	**George W.**	George W.	elect Barack	Barack Obama	George W.	2/3	0/3
ChronoGPT-Instruct$_{2020}$	**Bill Clinton**	**George W.**	**Barack Obama**	**Donald Trump**	Donald Trump	elect Mike	4/4	1/2
ChronoGPT-Instruct$_{1999}$ through ChronoGPT$_{2020}$							12/13	0/17

Table 2 Next-Token Predictions of U.S. Presidents using ChronoGPT-Instruct

This table presents ChronoGPT-Instruct's next-token predictions for prompts listing the incoming U.S. president alongside the three most recent predecessors. The model is tasked with predicting the name of the most recent president, which appears as the final missing entry in the sequence. The input prompt is structured as follows:

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

The table shows the predictions generated by each model in the ChronoGPT-Instruct series. Each prediction consists of exactly two tokens, selected deterministically by choosing the most probable token at each step. Gray shading indicates prompts referencing years beyond the model's knowledge cutoff, including elections in which the president-elect had not yet assumed office. Correct predictions are highlighted in blue. For comparison, outputs from GPT-2, GPT-2 XL (released in 2019), Llama-3.2-3B-Instruct (released in 2023), and Qwen-1.5-1.8B-Chat (released in 2024) are also included.

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
Llama-3.2-3B-Instruct	**scandal**, which	**SARS** (	**crisis**, which	**Brexit referendum**.	**COVID**-19	LLaMa	5/6	—
Qwen-1.5-1.8B-Chat	**scandal**, which	**SARS**.	**crisis**. The	**Brexit referendum**.	**COVID**-1	**GPT**-	6/6	—
ChronoGPT-Instruct$_{1999}$	ous Act that	the Guill	market, which	"vote of	the "Asia	the AI assistant	—	0/6
ChronoGPT-Instruct$_{2005}$	**scandal**, which	**SARS**,	market, which	"Voting	the Spanish flu	"AI-	2/2	0/4
ChronoGPT-Instruct$_{2010}$	**scandal**. It	the **severe acute**	**crisis**, which	"vote-	the Spanish flu	"The Computer	3/3	0/3
ChronoGPT-Instruct$_{2015}$	Corporation **scandal**,	**SARS**,	**crisis**, which	"UK Independence Ref	the Spanish flu	AI Language Model	3/3	0/3
ChronoGPT-Instruct$_{2020}$	**scandal**, which	**Severe Ac**	**crisis**, which	'**Brexit vote**.	**COVID**-	OpenAI,	5/5	0/1
ChronoGPT-Instruct$_{1999}$ through ChronoGPT-Instruct$_{2024}$							13/13	0/17

Table 3 Next-Token Predictions of Major Events using ChronoGPT-Instruct

This table presents ChronoGPT-Instruct's next-token predictions for prompts describing major historical events across a range of years. Panel A displays the input prompts for each event, and Panel B shows the corresponding predictions produced by each model in the ChronoGPT-Instruct series. Each prediction consists of exactly three tokens, selected deterministically by choosing the most probable token at each step. Gray shading indicates prompts that reference events beyond the model's knowledge cutoff. Correct predictions are highlighted in blue. For comparison, outputs from GPT-2, GPT-2 XL (released in 2019), Llama-3.2-3B-Instruct (released in 2023), and Qwen-1.5-1.8B-Chat (released in 2024) are also included.

Table 2 presents the test involving U.S. presidents and Table 3 presents the test involving major events. 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.

Within the respective knowledge window of each version of ChronoGPT-Instruct, the overall results are near-perfect (12 out of 13) for the U.S. presidents test and perfect (13 out of 13) for the major events test. The high accuracy in the pre-knowledge cutoff period reflects the quality and temporal relevance of the ChronoGPT-Instruct's knowledge. In contrast, during the post-cutoff period represented by the gray-shaded area, none of the ChronoGPT-Instruct models correctly predict any future president or major event. Overall, these findings validate that the textual data used to train or fine-tune our chronologically consistent models contains no evidence of leakage.

3.3 Prompt-Based Trading Portfolios

While He et al. (2025) highlight the potential of ChronoGPT to generate profitable signals through embeddings following Chen et al. (2023), it remains an open question how well directly prompting a chronologically consistent model can serve a similar purpose. Lopez-Lira and Tang (2023) demonstrate that prompting LLMs can yield robust trading signals for a limited time period after the model's knowledge cutoff, without introducing lookahead bias. With ChronoGPT-Instruct, we extend this line of inquiry by conducting prompt-based portfolio construction over a substantially longer time horizon, from January 2007 to July 2023.

To operationalize this investigation, we apply the following prompt to financial news headlines at the stock-day level, for all stocks with news coverage, similar to Lopez-Lira and Tang (2023):

```
{
    You are a financial expert with stock recommendation experience.  Answer
    GOOD if good news, BAD if bad news, or UNKNOWN if uncertain in the first
    line.  Then elaborate with one short and concise sentence on the next
    line.  {Is this headline / Are these headlines} GOOD or BAD for the stock
    price of {company} in the short term?
    ### Headline{s}:
    {headlines}
    ### Response:
}
```

We then form portfolios based on the LLM's response. The stock is assigned to the good news (G) or bad news (B) portfolio based on the first word generated by the LLM. If the first word is neither "good" nor "bad", the stock is assigned to the "unknown" portfolio. A

long-short portfolio ($G-B$) is formed from longing the good news portfolio and shorting the bad news portfolio.

We evaluate this strategy on ChronoGPT-Instruct$_{1999}$, which is free from lookahead bias, as the model is trained and tuned entirely on data before the sample period of January 2007–July 2023. For comparison, we also include results from Qwen-1.5-1.8B-Chat, Llama-3.2-3B-Instruct and Llama-3.2-1B-Instruct. The model closest in size is Qwen-1.5-1.8B-Chat, which has a 20% larger parameter count. The next larger model, Llama-3.2-3B-Instruct, has twice the parameter count. Both models are trained on substantially more data than ChronoGPT-Instruct.

	ChronoGPT-Instruct$_{1999}$			Qwen-1.5-1.8B-Chat		
	Mean	SD	SR	Mean	SD	SR
Bad news (B)	6.83	23.56	0.29	-65.48	57.03	-1.15
Unknown	6.37	23.95	0.27	7.38	23.61	0.31
Good news (G)	13.56	22.75	0.60	12.67	22.71	0.56
$G-B$	6.73	5.66	1.19	78.15	52.21	1.50
	Llama-3.2-3B-Instruct			Llama-3.2-1B-Instruct		
	Mean	SD	SR	Mean	SD	SR
Bad news (B)	-12.57	28.10	-0.45	-0.93	8.67	-0.11
Unknown	0.10	30.80	0.00	-3.58	30.26	-0.12
Good news (G)	10.01	23.00	0.44	9.35	23.05	0.41
$G-B$	22.58	13.95	1.62	10.28	23.53	0.44

Table 4 Performance of Prompt-Based Trading Portfolios

This table presents annualized performance metrics (mean return, standard deviation, and Sharpe ratio) for portfolios sorted by the LLM's direct response categorizing the news headlines as good news (G), bad news (B), or unknown. The $G-B$ row represents a strategy of going long on the portfolio of stocks classified as having good news (G) and short on those with bad news (B). All values are in percentage points except Sharpe ratios. All portfolios are equal-weighted and rebalanced daily. Data spans January 2007–July 2023.

Table 4 presents the results of the prompt-based trading portfolios. ChronoGPT-Instruct$_{1999}$, which is built on a 1.5B parameter model, achieves a Sharpe ratio of 1.19, outperforming Llama-3.2-1B-Instruct while underperforming the comparatively larger models, Qwen-

1.5-1.8B-Chat and Llama-3.2-3B-Instruct.

If a chronologically consistent model is comparable to an inconsistent model in every other aspect—constructed with the same parameter count and trained on the same amount of data—then an exact match in return performance would suggest that the inconsistent model's return predictability entirely persists even in the absence of leakage. Since ChronoGPT-Instruct is lower-capacity by design than Qwen-1.5-1.8B-Chat and Llama-3.2-3B-Instruct, we can use it to provide a conservative lower bound for quantifying lookahead bias in these models. Comparing ChronoGPT-Instruct$_{1999}$'s Sharpe ratio of 1.19 to that of Qwen-1.5-1.8B-Chat's (1.50) and Llama-3.2-3B-Instruct's (1.62), at least 73% to 79% of the apparent return predictability may persist even in the absence of data leakage. The remaining gap in Sharpe ratios (e.g., between 1.19 and 1.62) likely reflects a combination of differences in model capacity and lookahead bias in the comparison model.

Additionally, while the results for ChronoGPT-Instruct are modest, they highlight its viability for developing prompt-based trading strategies that avoid lookahead bias. Although improving performance is not our primary goal, these findings suggest that future scaling—if done while preserving chronological consistency—could further enhance its utility.

4 Conclusion

We address look-ahead bias in LLM-based prediction by releasing the first instruction-following language models whose training corpora are explicitly time-stamped. For example, ChronoGPT-Instruct$_{1999}$ is trained and subsequently instruction-fine-tuned exclusively on text available up to 1999, giving researchers over two decades of truly out-of-sample period for evaluation.

While we acknowledge that temporal constraints necessarily limit model performance

compared to contemporary alternatives, ChronoGPT-Instruct models offer a conservative lower bound for quantifying lookahead bias. A practical way to gauge the extent of look-ahead bias is to compare ChronoGPT-Instruct with similarly sized but temporally inconsistent models such as Qwen-1.5-1.8B-Chat or Llama-3.2-3B-Instruct. Our aim is thus not to achieve state-of-the-art performance, but to provide an easy-to-use, replicable benchmark for quantifying look-ahead bias in a wide range of prompt-based prediction tasks.

References

Chang, Anne, Xi Dong, Xiumin Martin, and Changyun Zhou, 2023, Ai democratization, return predictability, and trading inequality, *Available at SSRN 4543999* .

Chen, Jian, Guohao Tang, Guofu Zhou, and Wu Zhu, 2025, Chatgpt and deepseek: Can they predict the stock market and macroeconomy?, *arXiv preprint arXiv:2502.10008* .

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

Engelberg, Joseph, Asaf Manela, William Mullins, and Luka Vulicevic, 2025, Entity neutering, *Available at SSRN* .

Glasserman, Paul, and Caden Lin, 2023, Assessing look-ahead bias in stock return predictions generated by gpt sentiment analysis, *arXiv preprint arXiv:2309.17322* .

He, Songrun, Linying Lv, Asaf Manela, and Jimmy Wu, 2025, Chronologically consistent large language models, *arXiv preprint arXiv:2502.21206* .

Jha, Manish, Jialin Qian, Michael Weber, and Baozhong Yang, 2024, Chatgpt and corporate policies, Technical report, National Bureau of Economic Research.

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

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

Lv, Linying, 2025, Do sell-side analyst reports have investment value?, *arXiv preprint arXiv:2502.20489* .

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

Sarkar, Suproteem K, and Keyon Vafa, 2024, Lookahead bias in pretrained language models, *Available at SSRN* .