NBER WORKING PAPER SERIES

SCALING POINT-IN-TIME LANGUAGE MODELS

Bryan T. Kelly
Semyon Malamud
Johannes Schwab
Teng Andrea Xu

Working Paper 35247
http://www.nber.org/papers/w35247

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
May 2026

Bryan Kelly and Semyon Malamud are consultants at AQR Capital Management. Andrea Xu is an employee of AQR Capital Management. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

At least one co-author has disclosed additional relationships of potential relevance for this research. Further information is available online at http://www.nber.org/papers/w35247

NBER working papers are circulated for discussion and comment purposes. They have not been peer-reviewed or been subject to the review by the NBER Board of Directors that accompanies official NBER publications.

© 2026 by Bryan T. Kelly, Semyon Malamud, Johannes Schwab, and Teng Andrea Xu. All rights reserved. Short sections of text, not to exceed two paragraphs, may be quoted without explicit permission provided that full credit, including © notice, is given to the source.

Scaling Point-in-Time Language Models
Bryan T. Kelly, Semyon Malamud, Johannes Schwab, and Teng Andrea Xu
NBER Working Paper No. 35247
May 2026
JEL No. C14, C45, G11, G14, G17

ABSTRACT

Large language models trained on unrestricted internet corpora inevitably embed information from the future, introducing lookahead bias that compromises the validity of backtests and causal inference in finance and the social sciences. Point-in-time language models—trained exclusively on text available up to each calendar date—eliminate this leakage by construction, but existing efforts typically produce models that lag substantially behind their unconstrained counterparts. We show that this performance gap can be narrowed through scale. Training decoder-only transformers with up to 4 billion parameters on 1 trillion chronologically filtered tokens from FineWeb, we construct a sequence of monthly model checkpoints spanning 2013–2024. Across a range of common-sense reasoning and language understanding benchmarks, our models approach the performance of leading open-weight models of comparable size (such as Gemma-3-4B and LLaMA-7B) trained on temporally unrestricted data, although a performance gap remains on several tasks. Finally, in a strict out-of-sample economic evaluation task, portfolios built from point-in-time embeddings achieve robust positive Sharpe ratios and perform close to full-sample counterparts that violate temporal validity, indicating that chronologically consistent language models can extract economically meaningful signals without relying on look-ahead bias. We release the complete pipeline—including dataset construction, training infrastructure, and evaluation code—to enable reproducible point-in-time language modeling and to support research applications that require strict temporal validity.

Bryan T. Kelly
Yale University
and NBER
bryan.kelly@yale.edu

Johannes Schwab
École Polytechnique Fédérale
de Lausanne (EPFL)
johannes.schwab@epfl.ch

Semyon Malamud
Swiss Finance Institute
semyon.malamud@epfl.ch

Teng Andrea Xu
AQR Capital Management
andrea.xu@aqr.com

Scaling Point-in-Time Language Models

Bryan Kelly, Semyon Malamud, Johannes Schwab, and Teng Andrea Xu*

Abstract

Large language models trained on unrestricted internet corpora inevitably embed information from the future, introducing lookahead bias that compromises the validity of backtests and causal inference in finance and the social sciences. Point-in-time language models—trained exclusively on text available up to each calendar date—eliminate this leakage by construction, but existing efforts typically produce models that lag substantially behind their unconstrained counterparts. We show that this performance gap can be narrowed through scale. Training decoder-only transformers with up to 4 billion parameters on 1 trillion chronologically filtered tokens from FineWeb, we construct a sequence of monthly model checkpoints spanning 2013–2024. Across a range of common-sense reasoning and language understanding benchmarks, our models approach the performance of leading open-weight models of comparable size (such as Gemma-3-4B and LLaMA-7B) trained on temporally unrestricted data, although a performance gap remains on several tasks. Finally, in a strict out-of-sample economic evaluation task, portfolios built from point-in-time embeddings achieve robust positive Sharpe ratios and perform close to full-sample counterparts that violate temporal validity, indicating that chronologically consistent language models can extract economically meaningful signals without relying on look-ahead bias. We release the complete pipeline—including dataset construction, training infrastructure, and evaluation code—to enable reproducible point-in-time language modeling and to support research applications that require strict temporal validity.

Code is available at https://github.com/anonymouspitllm-collab/PIT-LLM.

1 Introduction

Large language models are increasingly used in economics, finance, and the social sciences to extract structured signals from unstructured text (Gentzkow et al., 2019; Korinek, 2023; Horton, 2023). These models enable new approaches to measuring expectations, analyzing narratives, and

*Bryan Kelly is at Yale School of Management, AQR Capital Management, and NBER; www.bryankellyacademic.org. Johannes Schwab is at the Swiss Finance Institute, EPFL. Semyon Malamud is at the Swiss Finance Institute, EPFL, and CEPR, and is a consultant to AQR. Teng Andrea Xu is at AQR Capital Management. Semyon Malamud gratefully acknowledges the financial support of the Swiss Finance Institute and the Swiss National Science Foundation, Grant 100018-228042. AQR Capital Management is a global investment management firm that may or may not apply similar investment techniques or methods of analysis as described herein. The views expressed here are those of the authors and not necessarily those of AQR. This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID lp46 on Alps. This paper was written with assistance from Claude, an AI assistant by Anthropic.

forecasting economic outcomes. However, current research largely relies on foundation models trained on temporally unrestricted data (Brown et al., 2020; Touvron et al., 2023; Team et al., 2025; Bi et al., 2024). Because these models are trained on the entire internet corpus simultaneously, they inevitably encode knowledge from future time periods relative to the historical data being analyzed. This phenomenon, commonly referred to as *lookahead bias* or training leakage (Glasserman and Lin, 2023; Lopez-Lira and Tang, 2023; Sarkar and Vafa, 2024; Levy, 2024; Ludwig et al., 2025; Huang et al., 2026), can invalidate empirical results. In financial applications, such bias can distort estimates of risk premia and invalidate backtests. In economic forecasting, it can lead to artificially optimistic predictions by implicitly incorporating future information.

A natural remedy is to train *point-in-time* language models: models whose training data are restricted to text published on or before a given date. Recent work has demonstrated the feasibility of this approach, first with (He et al., 2025a,b)(ChronoBERT and ChronoGPT), then followed by Yan et al. (2026) (DatedGPT), producing chronologically consistent models that outperform earlier no-leakage baselines. However, these models remain substantially smaller and weaker than leading open-weight models, raising a practical question: *how much performance must one sacrifice to eliminate lookahead bias in large language models?*

We show that the answer is: *very little*. By scaling both model size and training data—up to 4 billion parameters and 1 trillion tokens of chronologically filtered web text—we obtain point-in-time models whose common-sense reasoning and language comprehension approach those of Gemma-3-4B (Team et al., 2025) and LLaMA-7B (Touvron et al., 2023), models trained on full, temporally unrestricted corpora. Moreover, the economic forecasts produced by our point-in-time language models are free from lookahead and produce tradable portfolios with strong out-of-sample mean returns. Our results suggest that the performance gap attributed to temporal restrictions is largely a gap in scale (Henighan et al., 2020; Kaplan et al., 2020; Hoffmann et al., 2022), not an inherent limitation of the point-in-time paradigm.

Contributions. Our contributions are fourfold. *First*, we advance the state of the art in point-in-time LLM training. Starting from the GPT-2-based architecture of Jordan et al. (2024), we scale training data by approximately 140×, increase model size from 1.5B to 4B parameters, extend the context length from 1,536 to 2,048 tokens, and increase the embedding dimension from 768 to 4,096. The resulting models achieve zero-shot accuracy on standard benchmarks that is within a few percentage points of leading open-weight models trained without temporal constraints. *Second*, we instruction-tune our models using LoRA (Hu et al., 2022) and evaluate them on IFEval (Zhou et al., 2023), a programmatically verifiable benchmark that avoids the well-documented biases of LLM-as-a-judge evaluation (Zheng et al., 2024; Liu et al., 2023b; Shen et al., 2023; Wang et al., 2024). *Third*, we release the complete project resources—covering dataset construction, model training, evaluation code, and model checkpoints—thereby substantially lowering the barrier to reproducible point-in-time LLM research. *Fourth*, and foremost, we assess the economic value of our models in an asset pricing application. We construct point-in-time textual signals from news and use them to forecast returns and macroeconomic conditions. This setting provides a stringent out-of-sample test of whether temporally consistent language models extract economically meaningful information rather than inadvertently exploiting look-ahead bias. We show that point-

in-time models deliver robust predictive performance and economically significant gains relative to models trained on standard, non-temporally constrained corpora, highlighting the importance of chronological consistency for financial applications.

2 Literature Review

Text-based asset pricing. A large body of literature studies how news text forecasts stock returns. Early work relies on dictionary-based or word-count methods (Tetlock, 2007; Tetlock et al., 2008), while supervised approaches extract sentiment signals tailored to return predictions (Ke et al., 2019). More recently, large language models have enabled richer representations of news content. Chen et al. (2022) show that LLM embeddings of news articles strongly predict next-day returns, and Lopez-Lira and Tang (2023) demonstrate that prompting ChatGPT for headline sentiment yields robust trading signals. Bybee et al. (2024) use topic modeling on news text to construct interpretable macroeconomic factors. Didisheim et al. (2026) build on these advances by decomposing news embeddings into predictable and surprise components, uncovering a "pure news" anomaly that exceed many previously documented market anomalies. Crucially, they verify their findings using chronologically consistent LLMs from He et al. (2025a), confirming that the anomaly is not an artifact of lookahead bias—precisely the kind of application that motivates our work.

In-silico policy analysis. A long-standing objective in economics and public policy is the ability to conduct counterfactual experiments on complex social systems. In practice, real-world policy experiments are costly, slow, and often ethically constrained. As a result, policymakers frequently rely on observational data and structural modeling assumptions. Recent work suggests that large language models may serve as synthetic agents for conducting *in-silico experiments* that simulate the responses of individuals or institutions to hypothetical scenarios (Horton, 2023). However, the validity of such simulations depends critically on ensuring that models do not possess knowledge that would only become available in the future relative to the simulated time period. Point-in-time language models uniquely enable temporally consistent simulations. Policymakers could use these models to explore how populations might respond to policy changes, regulatory reforms, or economic shocks under realistic historical information environments.

Point-in-time language models. Our work builds on and extends the point-in-time LLM framework introduced by He et al. (2025a) and He et al. (2025b). In those papers, the authors train ChronoBERT and ChronoGPT, two families of chronologically consistent models with knowledge cutoffs beginning in 1999, and demonstrate competitive performance on NLP benchmarks and financial forecasting tasks. This line of work was recently extended in Yan et al. (2026). We demonstrate that this paradigm can be pushed substantially further through increased training data, model scaling, and cutting edge training techniques. Specifically, we scale up the original GPT-2-based ChronoGPT from a 1B-parameter model to a 4B-parameter decoder-only transformer trained on 1 trillion tokens, while extending the context length from 1,536 to 2,048 tokens and increasing the embedding dimension from 768 to 4,096 to align with architectures such as Mistral-7B (Jiang

et al., 2023). Because ChronoBERT is not a generative model, we use ChronoGPT and DatedGPT as the main benchmarks for the remainder of the paper.

3 Methodology

The current literature suggests that large language models should follow diffrent stage oftraining first be pre-trained on large-scale, general-purpose corpora to acquire broad linguistic and world knowledge (Brown et al., 2020; Touvron et al., 2023), and subsequently fine-tuned on instruction-following datasets to improve their generalization and zero-shot performance on unseen tasks (Wei et al., 2021; Sanh et al., 2022; Ouyang et al., 2022). In this work, we closely follow the curriculum training procedure described in Lambert et al. (2024), with a new initial stage devoted to producing a strong pre-trained point-in-time base model. Our primary motivation is twofold: first, to demonstrate that point-in-time models can achieve common-sense reasoning and language comprehension on par with models trained on full, temporally unrestricted corpora; and second, to provide a fully reproducible training pipeline, including code and data configurations, to facilitate future research.

Model. We train decoder-only large language models with 1.5B and 4B parameters (henceforth PIT-1.5B and PIT-4B, respectively), adapting the implementation of Jordan et al. (2024) to suit our requirements. The architecture builds upon the GPT framework (Radford et al., 2018) and incorporates several recent advances in optimization and scaling, including matrix function-based preconditioning (Higham, 2008; Schulz, 1933), learning rate modernization (Bernstein and Newhouse, 2024), distributed Shampoo optimization (Gupta et al., 2018; Anil et al., 2020), scaling strategies (Hägele et al., 2024), and architectural refinements such as value residual learning (Zhou et al., 2024), as demonstrated in Gemma 2 (Team et al., 2024). The model is trained using the standard next-token prediction objective (Radford et al., 2018). Under this autoregressive formulation, the joint probability of a sequence $(x_1, \ldots, x_T)$ factorizes as

$$p_\theta(x_1, \ldots, x_T) = \prod_{t=1}^{T} p_\theta(x_t \mid x_1, \ldots, x_{t-1}),$$

in practice, the model minimizes the negative log-likelihood (cross-entropy) of the next token:

$$\mathcal{L}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{t=1}^{T} \log p_\theta(x_t \mid x_1, \ldots, x_{t-1})\right].$$

The model is trained on a temporally ordered stream of tokens, directly aligning our setup with the incremental and continual learning literature (Gupta et al., 2023; Ke et al., 2023; Parmar et al., 2024; Chen et al., 2025). We checkpoint the model weights on a monthly basis. A complete description of the architecture and hyperparameters is provided in Appendix A.

Pre-Training. The selection of a pre-training corpus is a critical design decision, as an inappropriate data composition may lead to catastrophic forgetting (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017; Luo et al., 2025; Kotha et al., 2023). We use the FineWeb dataset (Penedo et al., 2024), which provides a high-quality, filtered and deduplicated snapshot of internet text spanning

from 2013 to 2025. The dataset comprises 15 trillion English tokens drawn from 96 Common Crawl (Common Crawl, 2024) snapshots, processed through language identification and quality filtering pipelines. Models trained on FineWeb have been shown to consistently outperform those trained on other publicly available web-scale corpora (Raffel et al., 2020a; Penedo et al., 2023; Soldaini et al., 2024; Soboleva et al., 2023; Ortiz Suárez et al., 2019; Gao et al., 2020) across a range of language understanding, reasoning, and knowledge benchmarks.

Instruction Fine-Tuning. Pre-training with a self-supervised next-token prediction objective on massive unlabeled corpora induces broad linguistic and world knowledge, but does not directly optimize performance on downstream tasks (Raffel et al., 2020b). Fine-tuning addresses this limitation and serves as a central mechanism of modern transfer learning in NLP, enabling pre-trained models to adapt to new domains, tasks, or instructions using relatively small amounts of labeled data while preserving the capabilities acquired during pre-training (Wei et al., 2021). In particular, instruction fine-tuning adapts a pre-trained model using a collection of datasets formatted as natural language instructions. Extensive work has demonstrated that this procedure consistently improves generalization to unseen tasks (Wei et al., 2022; Ouyang et al., 2022; Chung et al., 2022; Sanh et al., 2022; Lambert et al., 2024). While Lambert et al. (2024) perform a full-parameter fine-tune of the base model, we diverge from their approach and instead employ low-rank adaptation (LoRA) (Hu et al., 2022). In short, given a pre-trained weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA freezes W_0 and parameterizes the update as a low-rank decomposition

$$W = W_0 + BA, \qquad B \in \mathbb{R}^{d_{\text{out}} \times r}, \quad A \in \mathbb{R}^{r \times d_{\text{in}}}, \quad r \ll \min\{d_{\text{out}}, d_{\text{in}}\},$$

so that only the factors A and B are learned.[1] We adopt LoRA over full-parameter fine-tuning for three reasons. First, Biderman et al. (2024) demonstrate that LoRA acts as a strong implicit regularizer: it adapts the model to new tasks while incurring substantially less representational drift than full fine-tuning, thereby mitigating catastrophic forgetting of previously acquired capabilities. Second, LoRA is significantly more compute-efficient: with rank $r = 16$, the number of trainable parameters per layer reduces from $d_{\text{in}} \times d_{\text{out}}$ to $r(d_{\text{in}} + d_{\text{out}})$, a reduction of over two orders of magnitude for typical hidden dimensions. Finally, prior work shows that LoRA can match the performance of full-parameter fine-tuning within 1–2% when properly tuned, while preserving its primary advantages of reduced memory usage and faster training (Lee et al., 2026; Zhao et al., 2024).

Data. Table 1 summarizes the datasets used in each training stage. While the FineWeb corpus is already indexed with publication timestamps allowing us to enforce a temporal cutoff directly, the remaining fine-tuning datasets required additional curation to ensure that no instruction-response pairs reference real-world events beyond the target time horizon. We describe our temporal filtering procedure in Appendix B. In addition, we remove examples exceeding the model's context length and filter out non-English instances to maintain a consistent linguistic distribution across all training stages. We cap the Argilla/IFEval-like (Xu et al., 2024) data at 270,000 examples to maintain an approximate balance whereby half of the dataset comprises coding and mathematical problems, while the remaining half focuses on rigorous adherence to user instructions.

[1] We set $r = 16$ in all experiments.

Table 1

Datasets used across training stages. The FineWeb corpus is indexed by publication timestamp; all fine-tuning datasets are temporally filtered to remove references to real-world events beyond the target cutoff (Appendix B).

Stage	**Dataset** Penedo et al. (2024)	**Tokens**	**Filtered**
PT	`HuggingFaceFW/fineweb`	170B / 1T	–
Stage	**Dataset** Lambert et al. (2024); Xu et al. (2024)	**Examples**	**Filtered**
SFT	`ai2-adapt-dev/evol_codealpaca_heval_decontaminated`	106,790	100,114
	`ai2-adapt-dev/personahub_code_v2_34999`	34,943	34,748
	`ai2-adapt-dev/tulu_v3.9_open_math_2_gsm8k_50k`	50,000	49,772
	`ai2-adapt-dev/numinamath_tir_math_decontaminated`	64,191	63,910
	`ai2-adapt-dev/personahub_ifdata_manual_seed_v3_29980`	29,827	25,293
	`argilla/ifeval-like-data`	456,304	386,584

Evaluation. We rely on the widely used Gao et al. (2024) library to evaluate both our models and the benchmark models, supporting the reproducibility of our results. In short, Gao et al. (2024) provides the research community with a common library that offers standardized implementations for evaluating LLM performance across a wide range of end-to-end benchmarks.

4 Results

4.1 Common Sense Reasoning & Language Comprehension

Table 2

Zero-shot accuracy (%) on standard common sense reasoning benchmarks.

Model	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-easy	ARC-chal.	OBQA	Avg.
ChronoGPT_2024	60.4	66.5	43.9	54.9	52.5	29.5	34.8	48.9
DatedGPT_2024		70.5	53.2		52.0	34.7		52.6
PIT-1B_2024 (Ours)	61.9	76.1	64.3	59.4	49.5	30.4	34.8	53.8
PIT-4B_2024 (Ours)	**63.0**	**78.9**	**72.2**	**64.2**	**54.4**	**35.1**	**39.0**	**58.1**
Gemma-3-1B	66.4	74.8	62.0	58.9	72.2	38.3	37.0	58.5
Gemma-3-4B	79.0	80.0	76.0	69.5	81.8	54.9	43.0	69.2
LLaMA-7B	76.8	79.7	76.0	69.6	72.1	44.3	44.4	66.1

Following Touvron et al. (2023), we first evaluate our pre-trained models on common-sense reasoning tasks that assess whether language models have internalized broad world knowledge and can perform implicit reasoning beyond surface-level pattern matching. We report zero-shot accuracy on seven widely used benchmarks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC (Easy and Challenge) (Clark et al., 2018),

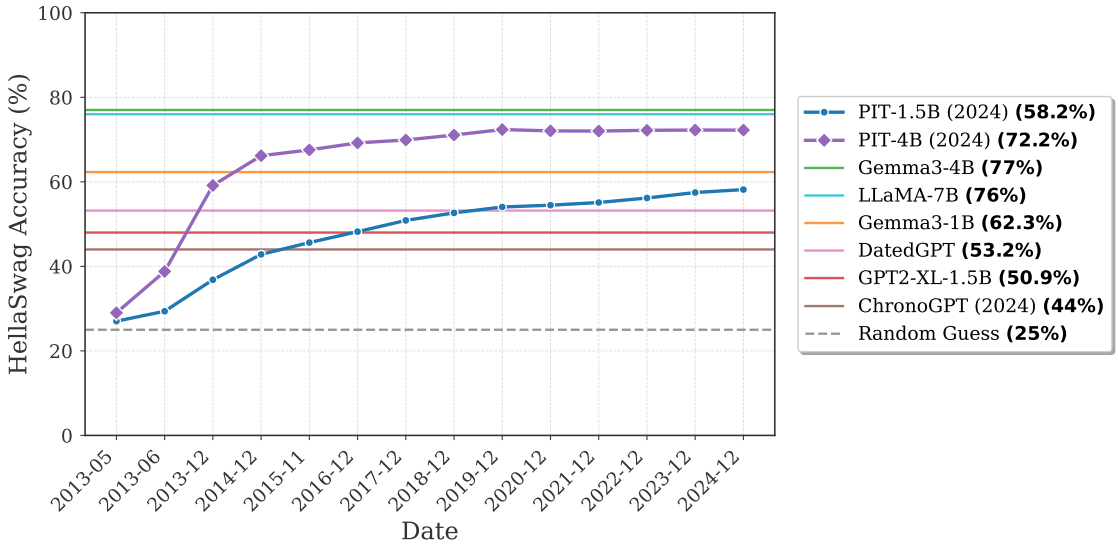

Figure 1: HellaSwag accuracy (%) over time for our PIT-1.5B (170B tokens) and PIT-4B (1T tokens) models trained on chronologically ordered FineWeb data. Horizontal lines denote published baselines: Gemma3-4B (77%) Team et al. (2025), Llama-7B (76%) Touvron et al. (2023), GPT2-XL (50.9%) Radford et al. (2019) (as reported in Wu et al. (2024)), Gemma3-1B (62.3%) Team et al. (2025), ChronoGPT 2024 (44%) He et al. (2025a), DateGPT (53.2%) Yan et al. (2026), and random guess (25%).

and OpenBookQA (Mihaylov et al., 2018). All evaluations are conducted in the zero-shot setting, following standard practice in the language modeling literature.

Table 2 reports the results.[2] PIT-4B outperforms every other point-in-time LLM, including its 1B-parameter counterpart PIT-1B_2024, ChronoGPT_2024 (He et al., 2025a), and DatedGPT Yan et al. (2026), with particularly large gains on HellaSwag (+28.3pp and +19pp over ChronoGPT_2024 and DatedGPT_2024, respectively). More importantly, PIT-4B closes much of the gap to LLaMA-7B and Gemma-3-4B—models trained without any temporal restriction and, in the case of LLaMA-7B, with nearly twice the parameter count. On PIQA and WinoGrande, the gap to LLaMA-7B and Gemma-3-4B is nearly negligible. These results demonstrate that chronological consistency need not come at the cost of strong language understanding.

4.2 Instruction Following

Evaluation with IFEval. A common approach to evaluating instruction-tuned models is to use an LLM as a judge (Liu et al., 2023a; Fu et al., 2023; Chiang and Lee, 2023). For instance, He et al. (2025b) evaluate their instruction-tuned ChronoGPT using AlpacaEval (Dubois et al., 2024), reporting length-controlled win rates against Qwen1.5-1.8B-Chat (Bai et al., 2023a). However, LLM-as-a-judge metrics are reference-free and suffer from well-documented biases: self-bias,

[2]At the time of writing, we were unable to locate the DatedGPT weights and therefore report the numbers from the original paper Yan et al. (2026).

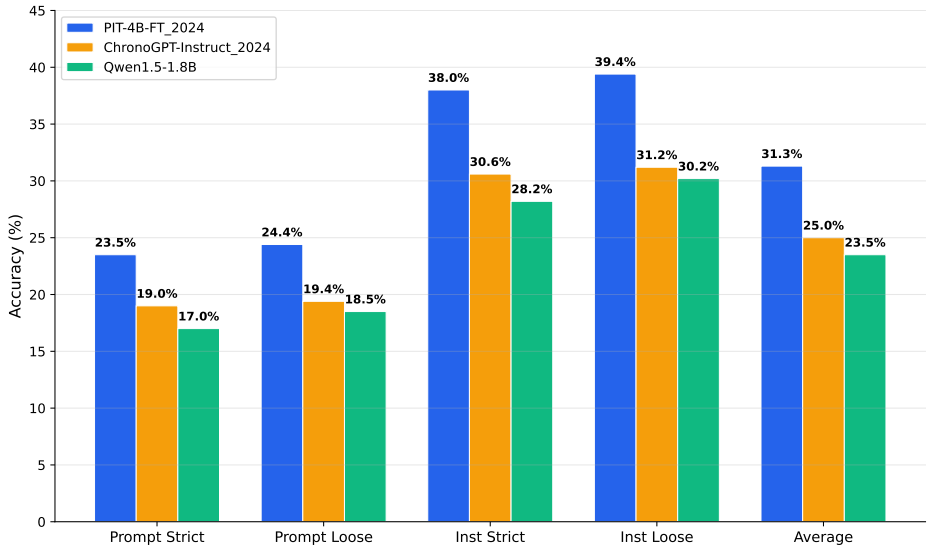

Figure 2: IFEval instruction-following accuracy (%) for our fine-tuned PIT-4B-FT_2024 compared with the point-in-time baseline ChronoGPT-Instruct_2024 He et al. (2025b) and the temporally unrestricted model Qwen1.5-1.8B Bai et al. (2023b).

where the judge prefers its own outputs (Liu et al., 2023b); positional bias, where the judge favors responses based on presentation order (Wang et al., 2023); and verbosity bias, where the judge prefers longer answers (Zheng et al., 2023). Zheng et al. (2024) demonstrate that a "null model" outputting constant, prompt-irrelevant responses can achieve top scores on AlpacaEval, Arena-Hard-Auto (Li et al., 2024), and MT-Bench (Bai et al., 2024). We refer the reader to Ye et al. (2024) for a comprehensive discussion of these failure modes.

To avoid these pitfalls, we evaluate instruction following using IFEval (Zhou et al., 2023), a benchmark that tests compliance with verifiable constraints—length limits, required keywords, formatting rules—checked programmatically rather than by an LLM judge. Figure 2 reports IFEval instruction-following accuracy for PIT-4B-FT, ChronoGPT-Instruct, and Qwen1.5-1.8B on four metrics: prompt-strict, prompt-loose, instruction-strict, and instruction-loose. PIT-4B-FT attains the best performance on all four metrics, averaging 31.3% versus 25.1% and 23.5% for ChronoGPT-Instruct and Qwen1.5-1.8B Bai et al. (2023b), respectively, showing that our larger, fine-tuned architecture delivers materially stronger instruction-following than both the smaller chronologically consistent baseline and Qwen trained on temporally unrestricted data.

Every model is evaluated using the default `IFEval` settings to maximize reproducibility, following the best practices of the public leaderboard[3]: temperature = 0, maximum generation tokens = 1280, and `do_sample=False`.

[3] https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard#/

5 The Economic Evaluation of Embeddings

A key motivation for point-in-time language models is their use in financial applications where lookahead bias can invalidate results. Following Chen et al. (2022) and Didisheim et al. (2026), we evaluate the economic value of our models by extracting news embeddings $E_t \in \mathbb{R}^{N_t \times d_h}$ and using them to predict stock returns $R_{t+1} \in \mathbb{R}^{N_t}$, where $N_t \in \mathbb{N}_+$ denotes the number of stocks available at time t and d_h is the embedding dimension. Because our models are trained exclusively on text available at each point in time, any return predictability we document is free of training leakage by construction, *meaning the portfolios conditioned on stock news are fully tradable.*

To quantify economic performance, we evaluate portfolios using the Sharpe ratio (SR) Sharpe et al. (1998), the standard economic metric of risk-adjusted return, which measures average return per unit of volatility. This provides a natural analogue to downstream evaluation in machine learning: a model is useful insofar as it produces signals that translate into high SR portfolios out-of-sample.

Data and Point-in-Time implementation. We use the Dow Jones Newswire dataset which spans the period from June 1979 to May 2020. Because the earliest available point-in-time (PIT) language model is timestamped December 2013, we report out-of-sample performance for 2014 through May 2020 only. Starting in 2014, we implement a strictly out-of-sample embedding procedure: for each calendar year t, we use the PIT model available at the end of the previous year (December $t-1$) to embed all articles published during year t. For example, the December 2013 model is used for 2014 articles, the December 2014 model for 2015 articles, and so on. This rolling scheme ensures that embeddings are constructed using only information available at each point in time, thereby avoiding look-ahead bias.

Full-sample benchmark models. In addition to the point-in-time (PIT) specifications, we consider two benchmark models based on the final checkpoints of our PIT models, denoted *PIT-4B-full* and *PIT-4B-FT-full*. Concretely, these correspond to the last available checkpoints obtained after training on the full sample. These models are fixed over time and therefore incorporate information from the entire sample, including future observations relative to each evaluation date. Consequently, they do not satisfy the strict temporal constraints imposed on the PIT models.

Embedding Construction. Let d_h denote the dimension of the final hidden representation of the language model (e.g., $d_h = 4096$ for PIT-4B). For an article i about stock s published on day d, let

$$\mathbf{h}_{d,s}^{(i)} \in \mathbb{R}^{d_h}$$

be the final-layer hidden state of the last token. We define the article embedding as

$$\mathbf{e}_{d,s}^{(i)} := \mathbf{h}_{d,s}^{(i)}.$$

We aggregate embeddings in two steps. First, we construct daily stock-level embeddings:

$$\bar{\mathbf{e}}_{d,s} = \frac{1}{N_{d,s}} \sum_{i=1}^{N_{d,s}} \mathbf{e}_{d,s}^{(i)},$$

where $N_{d,s}$ is the number of articles about stock s on day d. Second, we aggregate to the monthly level:

$$\tilde{\mathbf{e}}_{t,s} = \frac{1}{|\mathcal{D}_t|} \sum_{d \in \mathcal{D}_t} \bar{\mathbf{e}}_{d,s},$$

where $\mathcal{D}_t$ denotes the set of trading days in month t.

News Only Information & Anisotropy. To isolate the component of news embeddings that is orthogonal to well-known cross-sectional predictors, we residualize $\tilde{\mathbf{e}}_{t,s}$ with respect to a rich set of firm characteristics. Specifically, we use the characteristics compiled by Jensen et al. (2023) (JKP), denoted $S_{t,s} \in \mathbb{R}^K$, which summarize standard risk factors and anomalies widely used in the asset pricing literature, and we apply the same data pre-processing as in Didisheim et al. (2024). Then, for each month t, we run the cross-sectional regression

$$\tilde{\mathbf{e}}_{t,s} = \alpha + \beta S_{t,s} + \hat{\boldsymbol{\epsilon}}_{t,s}, \qquad \beta \in \mathbb{R}^{d_h \times K}, \qquad \alpha \in \mathbb{R}^{d_h}$$

and use the residual $\hat{\boldsymbol{\epsilon}}_{t,s}$ as the characteristic-adjusted embedding. Thus, we work with monthly observations $E_t = [\hat{\epsilon}_{t,1}, \ldots, \hat{\epsilon}_{t,N_t}]^\top \in \mathbb{R}^{N_t \times d_h}$. It is worth noting that this process not only extracts "pure" news information but also solves the well-known anisotropy problem that generative model embeddings usually suffer from; see Ethayarajh (2019); Gao et al. (2019).

Portfolio Construction. We construct d_h base portfolios from the embeddings and denote their returns at time t by

$$\boldsymbol{F}_t = (F_{t,1}, \ldots, F_{t,d_h})^\top \in \mathbb{R}^{d_h}, \quad \text{where} \quad F_{t,i} = E_{t-1,i}^\top R_t \in \mathbb{R}.$$

Thus, the embedding matrix maps stock-level returns into d_h base portfolio returns, one for each embedding coordinate. We then combine these base portfolios using penalized Maximum Sharpe Ratio Regression (MSRR) Kelly and Xiu (2023). This step can be interpreted as a regularized mean-variance optimization over the span of the base portfolios, in the spirit of Markowitz mean-variance portfolio choice (Markowitz, 1952) and the regression representation of the tangency portfolio in Britten-Jones (1999). We estimate MSRR weights using an expanding window $\mathcal{T}$ with an initial estimation period of 12 months.

$$\hat{\boldsymbol{\lambda}}_t(z) = \arg\min_{\boldsymbol{\lambda} \in \mathbb{R}^{d_h}} \frac{1}{\mathcal{T}} \sum_{u=1}^{\mathcal{T}} \left(1 - \boldsymbol{\lambda}^\top \mathbf{F}_u\right)^2 + z\|\boldsymbol{\lambda}\|_2^2. \tag{1}$$

The resulting tradable portfolio is

$$\boldsymbol{\pi}_t^{\text{PIT}}(z) = E_t \hat{\boldsymbol{\lambda}}_t(z) \in \mathbb{R}^{N_t}.$$

and the portfolio return is

$$r_{t+1}^{\text{PIT}} = \boldsymbol{\pi}_t^{\text{PIT}}(z)^\top R_{t+1}.$$

Hence, the final PIT portfolio is obtained by first constructing embedding-sorted base portfolios and then choosing the maximum-Sharpe combination of those portfolios. At each rebalancing date,

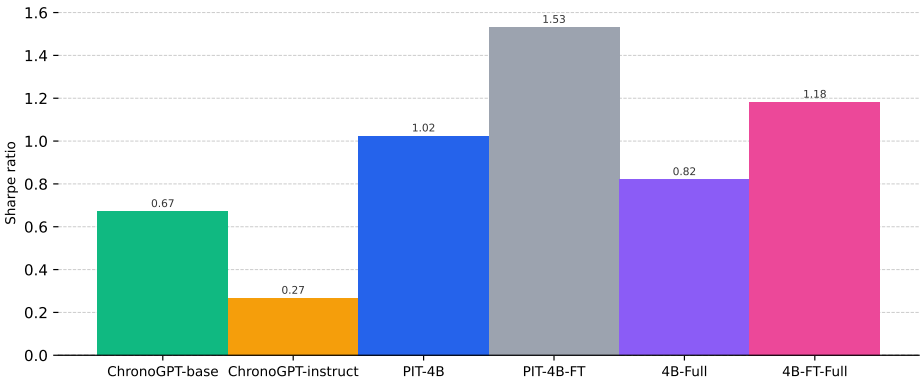

Figure 3: Out-of-sample annualized Sharpe ratio for each model.

we solve the MSRR problem over a grid of shrinkage parameters $z \in \mathcal{Z}$. We rescale each z according to

$$z_{\text{eff}} = \frac{d_h}{\mathcal{T}}\, z \cdot \frac{1}{d_h} \operatorname{tr}\left(\frac{1}{\mathcal{T}} \dot{\mathbf{F}}_{\mathcal{T}}^{\top} \dot{\mathbf{F}}_{\mathcal{T}}\right), \qquad \mathcal{Z} \in \{10^{-6}, \ldots, 10^{-1}, 1, 2, 5, 10, 10^{2}\}$$

where $\dot{\mathbf{F}}_{\mathcal{T}} = (\boldsymbol{F}_1, \ldots, \boldsymbol{F}_{\mathcal{T}})^{\top} \in \mathbb{R}^{\mathcal{T} \times d_h}$, and $\operatorname{tr}(\dot{\mathbf{F}}_{\mathcal{T}}^{\top} \dot{\mathbf{F}}_{\mathcal{T}} / \mathcal{T})$ is the total second moment of the base portfolio returns estimated over the rolling window Chernov et al. (2025).

For each value of z, we compute the returns of the strategy. We then rescale them to have a target volatility of 10% and build an equally weighted portfolio of the strategy across the grid of z. All performance statistics are computed over the post-December 2013 period, which constitutes the true out-of-sample evaluation.

Figure 3 reports out-of-sample annualized Sharpe ratios of the MSRR portfolios across model variants. Three main findings emerge:

1. **Point-in-time models produce economically meaningful signals** even under strict temporal constraints. PIT models deliver consistently positive Sharpe ratios, substantially outperforming smaller ChronoGPT variants, which often exhibit weaker performance. Importantly, their performance remains close to that of the corresponding full-sample benchmarks, indicating that enforcing strict temporal validity does not materially reduce economic significance.

2. **Scaling substantially improves performance**. Moving from smaller ChronoGPT models to PIT-4B variants leads to large and systematic increases in Sharpe ratios. For example, Sharpe ratios increase from roughly 0.7 for ChronoGPT-base to around 1.0–1.5 for PIT-4B models. This indicates that model scale plays a central role in extracting economically valuable signals.

3. **Fine-tuning further enhances performance at scale**. The fine-tuned PIT-4B model (PIT-4B-FT) outperforms its base counterpart across all size groups. This suggests that fine-tuning might improve robustness and generalization of the extracted signals across the cross-section.

Overall, these results provide strong evidence that strict temporal validity does not eliminate economically useful information, and that both scaling and fine-tuning play key roles in recovering and amplifying that information.

6 Conclusion

This paper examines whether enforcing strict temporal validity in language model training requires sacrificing either general language-model performance or the downstream economic value of text embeddings. Our results suggest that it does not. By scaling point-in-time pre-training to 1 trillion chronologically filtered tokens and 4 billion parameters, we obtain models that substantially outperform prior chronologically consistent baselines and approach the performance of strong open-weight models trained on temporally unrestricted corpora. The main message is that much of the apparent cost of eliminating lookahead bias reflects a scale deficit rather than an inherent limitation of the point-in-time paradigm.

We show this both in standard NLP evaluations and in a financially meaningful downstream application. On common-sense reasoning and language comprehension benchmarks, PIT-4B closes a large share of the gap to Gemma-3-4B and LLaMA-7B despite operating under a strict chronological constraint. After instruction fine-tuning with LoRA on temporally filtered data, the model also exhibits materially improved instruction-following performance on IFEval, indicating that temporal consistency can be preserved throughout the post-training pipeline. In an asset-pricing application, embeddings extracted from our point-in-time models generate economically meaningful out-of-sample return predictability, with larger models delivering stronger Sharpe ratios. These findings highlight the practical value of point-in-time models in empirical settings where leakage can otherwise invalidate inference.

More broadly, our results suggest that point-in-time language modeling is now a viable foundation for research in finance, economics, and the social sciences. In many of these settings, real-world policy interventions are costly to implement, slow to evaluate, politically constrained, and often ethically or administratively infeasible to randomize. Historically grounded in silico experiments can therefore play an important role in the design of complex social systems. But such exercises are credible only when the model's information set respects chronology: once temporal leakage is present, ex post information contaminates the historical decision environment and undermines both causal and counterfactual interpretation. Point-in-time LLMs provide a natural computational remedy by enforcing chronological consistency in the training corpus. Our results show that researchers no longer need to choose between temporal validity and competitive language-model performance. By releasing the full training, filtering, and evaluation pipeline, we aim to make chronologically consistent language modeling reproducible and easier to adopt in applications where causal interpretation, historical fidelity, and backtest integrity are essential.

Limitations. Several limitations remain. First, although scaling greatly narrows the gap to unrestricted models, it does not eliminate it completely, especially on some reasoning-intensive tasks. Second, our asset-pricing application focuses on a single benchmark dataset and a single

portfolio construction framework; broader evidence across tasks and domains remains an important direction for future work. Third, while our fine-tuning data are aggressively filtered to mitigate temporal leakage, stronger methods for temporally robust post-training and preference alignment warrant further study. Finally, our current evaluation is incomplete for the most recent period, and updated results for 2022–2025 will be reported in a subsequent version.

Overall, the evidence in this paper points to a simple conclusion: point-in-time LLMs can be made both credible and useful at a modern scale. This opens the door to a new generation of temporally grounded models for scientific measurement, historical analysis, and decision-making under genuine information constraints.

References

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. *arXiv:2002.09018*, 2020.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, et al. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023a.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv:2309.16609*, 2023b.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv:2409.20325*, 2024.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv:2401.02954*, 2024.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv:2405.09673*, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

Mark Britten-Jones. The sampling error in estimates of mean-variance efficient portfolio weights. *The Journal of Finance*, 54(2):655–671, 1999.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Leland Bybee, Bryan Kelly, Asaf Manela, and Dacheng Xiu. Business news and business cycles. *The Journal of Finance*, 79(5):3105–3147, 2024.

Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, et al. Towards effective and efficient continual pre-training of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5779–5795, 2025.

Yifei Chen, Bryan T Kelly, and Dacheng Xiu. Expected returns and large language models. *Available at SSRN 4416687*, 2022.

Mikhail Chernov, Bryan T Kelly, Semyon Malamud, and Johannes Schwab. A test of the efficiency of a given portfolio in high dimensions. Technical report, NBER:33565, 2025.

Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? *arXiv:2305.01937*, 2023.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *arXiv:2210.11416*, 2022.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *NAACL-HLT*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. Think you have solved question answering? try arc, the ai2 reasoning challenge. In *arXiv:1803.05457*, 2018.

Common Crawl. Common crawl corpus. https://commoncrawl.org, 2024. Accessed: January, 2026.

Antoine Didisheim, Shikun Barry Ke, Bryan T Kelly, and Semyon Malamud. Apt or "aipt"? the surprising dominance of large factor models. Technical report, National Bureau of Economic Research, 2024.

Antoine Didisheim, Bryan T. Kelly, Mo Pourmohammadi, and Hanqing Tian. The inefficient pricing of news. 2026. SSRN:6540399.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv:2404.04475*, 2024.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. pages 55–65. Association for Computational Linguistics, 2019.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. *arXiv:2302.04166*, 2023.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *arXiv:1907.12009*, 2019.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *arXiv:2101.00027*, 2020.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

Matthew Gentzkow, Bryan Kelly, and Matt Taddy. Text as data. *Journal of Economic Literature*, 57(3):535–574, 2019.

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

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to (re) warm your model? *arXiv:2308.04014*, 2023.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1842–1850. PMLR, 2018.

Alexander Hägele et al. Scaling laws and compute-optimal training beyond fixed training durations. *arXiv:2405.18392*, 2024.

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

Songrun He, Linying Lv, Asaf Manela, and Jimmy Wu. Instruction tuning chronologically consistent language models. *arXiv:2510.11677*, 2025b.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv:2010.14701*, 2020.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, 2008.

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

John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3, 2022.

Wenqian Huang, Albert J. Menkveld, and Shihao Yu. AI "errors". *Working Paper*, 2026. SSRN 6408138.

Theis Ingerslev Jensen, Bryan Kelly, and Lasse Heje Pedersen. Is there a replication crisis in finance? *The Journal of Finance*, 78(5):2465–2518, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

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

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

Zheng Tracy Ke, Bryan T Kelly, and Dacheng Xiu. Predicting returns with text data. *NBER:26186*, 2019.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. *arXiv:2302.03241*, 2023.

Bryan Kelly and Dacheng Xiu. Financial machine learning. *Foundations and Trends® in Finance*, 13(3-4):205–363, 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Anton Korinek. Language models and cognitive automation for economic research. Technical report, national Bureau of economic Research, 2023.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. *arXiv:2309.10105*, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv:2411.15124*, 2024.

Yu-Ang Lee, Ching-Yun Ko, Pin-Yu Chen, and Mi-Yen Yeh. Learning rate matters: Vanilla lora may suffice for llm fine-tuning. *arXiv:2602.04998*, 2026.

Bradford Levy. Caution ahead: Numerical reasoning and look-ahead bias in AI models. *Fama-Miller Working Paper*, 2024. SSRN 5082861.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv:2406.11939*, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv:2303.16634*, 2023a.

Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. Llms as narcissistic evaluators: When ego inflates evaluation scores. *arXiv:2311.09766*, 2023b.

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

Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Large language models: An applied econometric framework. *NBER:33344*, 2025.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Harry Markowitz. Modern portfolio theory. *Journal of Finance*, 7(11):77–91, 1952.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on empirical methods in natural language processing*, 2018.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache, 2019.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv:2203.02155*, 2022.

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models. *arXiv:2407.07263*, 2024.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67, 2020a.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020b.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 2021.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. *arXiv:2110.08207*, 2022.

Suproteem K Sarkar and Keyon Vafa. Lookahead bias in pretrained language models. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2024.

Günther Schulz. Iterative berechnung der reziproken matrix. *Zeitschrift für Angewandte Mathematik und Mechanik*, 13:57–59, 1933.

William F Sharpe et al. The sharpe ratio. *Streetwise–the Best of the Journal of Portfolio Management*, 3(3):169–85, 1998.

Chenhui Shen, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. Large language models are not yet human-level evaluators for abstractive summarization. In *Conference on empirical methods in natural language processing*, pages 4215–4233, 2023.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R. Steeves, Joel Hestness, and Nolan Dey. Slimpajama: A 627b token cleaned and deduplicated version of redpajama. https://www.cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, June 2023. Accessed: 2026-02-24.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788. Association for Computational Linguistics, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv:2503.19786*, 2025.

Gemma Team et al. Gemma 2: Improving open language models at a practical size. *arXiv:2408.00118*, 2024.

Paul C Tetlock. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168, 2007.

Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. More than words: Quantifying language to measure firms' fundamentals. *The journal of finance*, 63(3):1437–1467, 2008.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *arXiv:2305.17926*, 2023.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, 2024.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv:2109.01652*, 2021.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *arXiv:2109.01652*, 2022.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 944–964, 2024.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv:2406.08464*, 2024.

Yutong Yan, Raphael Tang, Zhenyu Gao, Wenxi Jiang, and Yao Lu. Datedgpt: Preventing lookahead bias in large language models with time-aware pretraining. *arXiv:2603.11838*, 2026.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. Justice or prejudice? quantifying biases in llm-as-a-judge. *arXiv:2410.02736*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv:1905.07830*, 2019.

Justin Zhao, Timothy Wang, Wael Abid, Geoffrey Angus, Arnav Garg, Jeffery Kinnison, Alex Sherstinsky, Piero Molino, Travis Addair, and Devvret Rishi. Lora land: 310 fine-tuned llms that rival gpt-4, a technical report. *arXiv:2405.00732*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Cheating automatic llm benchmarks: Null models achieve high win rates. *arXiv:2410.07137*, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv:2311.07911*, 2023.

Zhanchao Zhou et al. Value residual learning for alleviating attention concentration in transformers. *arXiv:2410.17897*, 2024.

A Training Configuration

We train two GPT-2-style decoder-only transformer models on chronologically ordered FineWeb data. PIT-1.5B has 52 layers, 12 attention heads, and an embedding dimension of 1,536 (head dim = 128), totaling approximately 1.5B parameters trained on 170B tokens. PIT-4B has 20 layers, 32 attention heads, and an embedding dimension of 4,096 (head dim = 128), totaling approximately 4.2B parameters trained on 1T tokens. Both models use a vocabulary of 50,304 tokens with a sequence length of 2,048. Training employs a hybrid optimizer: Muon (momentum = 0.95) for the transformer blocks and AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$) for the language model head, with a peak learning rate of 0.0009 and no weight decay. The learning rate follows a trapezoidal schedule with linear warmdown over the final 50% of training steps. We ran all experiments on NVIDIA GH200s and H100s, depending on queue and resource availability. PIT-4B achieved roughly 10,000 tokens per second per GPU, so the full pretraining run took roughly 28,000 GPU-hours.

B Removing Temporal Leakage

Similar to He et al. (2025b), we employ gpt-5-nano to identify and remove temporally sensitive examples from the datasets used in the supervised fine-tuning stage. We present each example from the SFT and DPO datasets to gpt-5-nano, prepending the prompt illustrated in Figure 4, and retain only those examples classified as "timeless"—namely, examples that do not reference real-world events, real individuals, specific dates, pop culture, or trending topics. This filtering step is designed to ensure that the training data remains invariant to temporal context, thereby preventing the model from internalizing ephemeral or potentially outdated knowledge during fine-tuning. Notably, we observe that the majority of examples in both datasets are naturally retained through this process, as instruction-following data predominantly comprises verifiable format constraints rather than world knowledge (Zhou et al., 2023), while mathematical and coding problems are inherently grounded in abstract reasoning, formal logic, and counterfactual scenarios that are not subject to temporal drift.

```
You are a binary classifier.  Determine if text contains TIME-SENSITIVE FACTS that
could become outdated or incorrect over time.

Output ONLY "0" (timeless) or "1" (time-aware).

Output "1" ONLY if the text states facts that could BECOME OUTDATED, such as:
-- Who currently holds a political office ("Biden is the president")
-- Recent or dated events ("The 2024 Olympics were held in Paris")
-- Current statistics, prices, or rankings ("Tesla stock is at $X")
-- A real person doing something specific at a specific time ("Elon Musk announced
X in 2024")
-- Laws, policies, or regulations tied to a specific time frame

Output "0" if the text is:
-- Math problems or solutions (even with character names like "Alice" or "John")
-- Programming/coding tasks or tutorials (even mentioning real tools:  Python,
Java, NetBeans, Eclipse, AWS, Docker, etc.)
-- General educational content about real software, frameworks, or technologies
-- Generic advice or best practices (even about real products)
-- Hypothetical scenarios (even with human names)
-- Instruction-following tasks
-- Scientific facts that don't change ("water boils at 100C")
-- Abstract reasoning or logic puzzles

KEY DISTINCTION: Mentioning a real tool, company, or product by name does NOT
make text time-aware.  Only FACTUAL CLAIMS THAT COULD BECOME OUTDATED do.

Examples:
-- "Use NetBeans profiler for CPU analysis" → 0 (generic advice about a tool)
-- "Write a REST API using Flask and AWS Lambda" → 0 (coding tutorial)
-- "Google announced Gemini 2.0 in December 2024" → 1 (dated event)
-- "The president of the United States is Joe Biden" → 1 (will become outdated)
-- "Tesla's market cap exceeded $1 trillion in 2024" → 1 (time-sensitive fact)
-- "Solve:  2x + 3 = 7" → 0 (math)
-- "Anna bought 5 apples at $2 each" → 0 (hypothetical)
-- "Python's GIL prevents true multithreading" → 0 (technical fact, stable)
-- "React 18 introduced concurrent rendering" → 0 (historical tech fact, won't
change)
-- "As of 2024, React is the most popular framework" → 1 (ranking changes over
time)
```

Figure 4: Prompt used for temporal leakage classification.