LLM EVALUATION IS A FORECASTING PROBLEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard procedure in machine learning is to randomly split train and test data, emulating IID draws from one underlying data distribution. In practice, models are trained on past data and deployed in the future, which may come from distinct distributions. In this paper, we observe that text distributions shift continuously over time and thus, because the goal of model evaluation is to estimate how well a model will perform in the future, we argue that evaluation should reflect the temporal structure of the data. Indeed, we find that web text is highly non-stationary, as documents grow more compressible and semantics, domains, and embeddings shift over time. We further observe that these shifts are temporally local, and model performance changes continuously over time across diverse benchmarks; consequently, future model performance is better predicted by measurements made on recent evaluation data, and is best predicted by time-series forecasters applied to historic performance rather than by historic performance itself. Choosing which model to deploy based on forecasted accuracy enables more reliable selection of performant models, thus improving downstream accuracy in the regime practitioners care about: deployment. Taken together, our experiments suggest that time should be treated as a fundamental variable in LLM evaluation.

1 INTRODUCTION

Among the first lessons a machine learning student learns is to randomly split their dataset into train and validation splits. This convention is well-motivated by *exchangeability*: if we assume that samples are independent draws from the same distribution, loss computed on a held-out validation set provides an unbiased estimate of expected loss on new samples during deployment as long as they are drawn from the same distribution (Shalev-Shwartz & Ben-David, 2014).

The importance of minimizing distribution shift between evaluation and deployment predates modern machine learning benchmarks. In an early NIST handwriting-recognition competition, the training data were written by Census workers, whereas the official test data were collected from high-school students; systems achieving less than 1% error on validation sets drawn from the training distribution performed substantially worse on the official test set. LeCun (1995) subsequently repartitioned the NIST data when constructing MNIST, splitting examples from both source populations across training and test sets while keeping writers disjoint to emulate identically distributed samples.

Large language models (LLMs) violate the assumption that their training, evaluation, and deployment samples are drawn from the same distribution. There may be a wide variety of distribution shifts from training to deployment, and frontier labs attempt to close these gaps by improving their training and evaluation coverage. One difficulty in closing these gaps is that the data from training and deployment differ along the temporal axis. LLM training data is gathered from historic web scrapes, and a large portion of the data is born years before training commences (Brown et al., 2020; Gao et al., 2020; Lazaridou et al., 2021). In contrast, data seen during deployment is often produced after the training cutoff. In this paper, we investigate the extent to which the temporal distribution shift affects how we should be evaluating LLMs and, by treating evaluation as a forecasting problem, propose best practices for mitigating this shift and improving estimates of model quality.

We begin by observing that text distributions are highly nonstationary under a wide variety of statistical metrics and in a manner that heavily influences LLM training and evaluation. The nonstationarity of text data distributions makes LLM evaluation closer to *time series forecasting* than to classical i.i.d. estimation. Inspired by *backtesting* from the financial engineering community (White, 2000;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

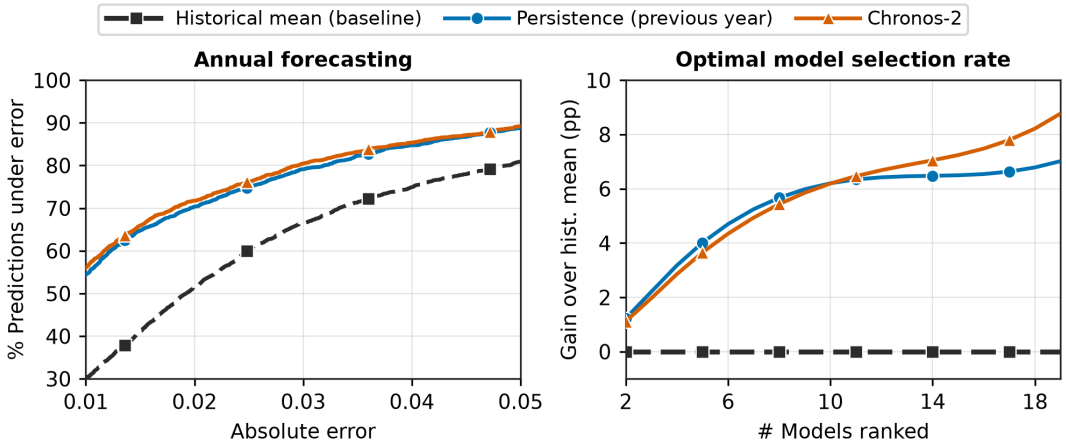

Figure 1: **Recent evaluations and time-series forecasting outperform temporally pooled benchmark scores for both future performance prediction and model selection.** We evaluate 19 open-source models on 12 temporally labeled datasets, using the historical mean as the pooled baseline. *(Top)* The most recent year and Chronos-2 yield fewer predictions exceeding each error threshold. *(Bottom)* Both improve annual model selection, with larger gains as the candidate set grows.

Bailey et al., 2017), we treat performance as a time series and ask how well measurements available at one point in time predict performance on later data. Even though randomly shuffling data across dates erases precisely the dependence that is useful for predicting future performance, standard LLM benchmarks commonly collapse examples across many dates into one score, including old samples which poorly predict future performance and obscure temporal structure.

Instead, we propose applying forecasting algorithms, like Chronos-2 (Ansari et al., 2025) or Simple Exponential Smoothing (SES) (Holt, 1957), to the time-series of historic model accuracy computed on data from different months or years in order to better predict future performance than historic model accuracy itself. Indeed, Figure 1 demonstrates the efficacy of this approach in reducing prediction error relative to standard practice. While improved forecasting is interesting in its own right, one of its most important applications is *model selection*: given a task and several models, which one should we choose to deploy? We find that the forecasting methods that leverage temporal information lead to substantially better deployment decisions than the temporally agnostic baseline, demonstrating the concrete benefits of treating evaluation as a forecasting problem.

We discuss related work in Section A. Our contributions are summarized as follows:

- **We observe that web text is strongly non-stationary over time.** Across three popular pretraining corpora, lexical, topical, syntactic, and semantic properties shift systematically, and models generalize best between temporally adjacent train and test data.
- **We show that evaluation staleness matters.** Across 19 open-source models and 12 benchmarks, recent evaluations consistently predict near-future model performance better than older ones, with stale evaluations producing larger tail errors.
- **We frame LLM evaluation as a forecasting problem.** Time-aware predictors, such as simple persistence and Chronos-2, outperform the standard practice of averaging all benchmark scores for predicting future model performance.
- **We show that better forecasts improve model selection for deployment.** Time-aware evaluation increases the rate of selecting the best future-performing model and reduces the frequency of high-regret model choices.

2 Temporal Shifts in Pretraining Data

We first investigate temporal distributional shifts in pretraining datasets. In Section 2.1, we show that pretraining corpora derived from Common Crawl are **non-stationary** by analyzing lexical metrics, domain shift, and embedding drift. Then, in Section 2.2, we show that model generalization

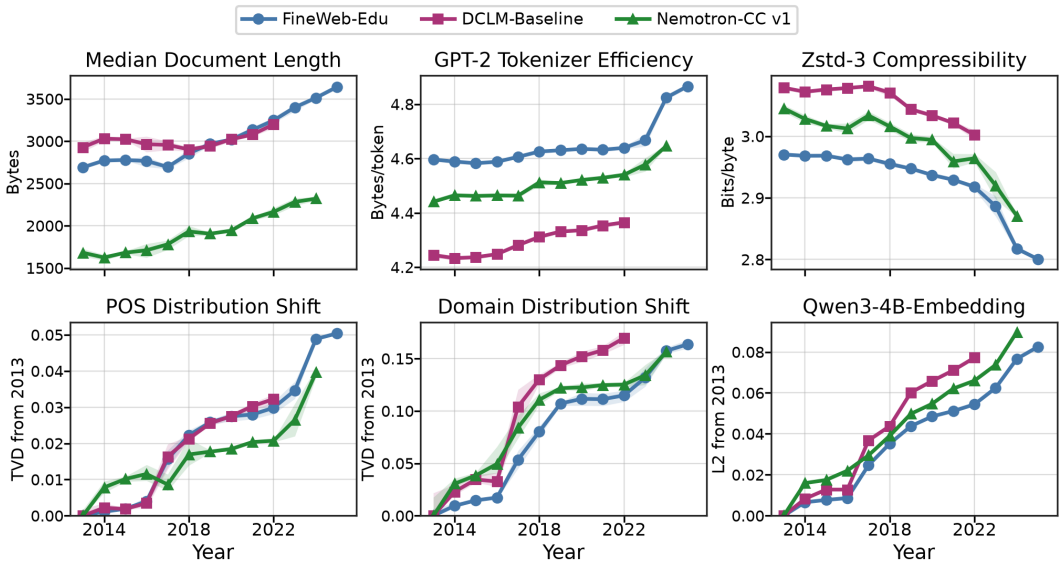

Figure 2: **Surface-level, syntactic, topical, and semantic properties of pretraining corpora systematically change over time.** Across FineWeb-Edu, DCLM-Baseline, and Nemotron-CC v1, document length and tokenizer efficiency increase while byte compressed-size ratio decreases. Part-of-speech and domain distributions, along with annual document-embedding centroids, increasingly diverge from their initial values. Shading shows bootstrapped 95% confidence intervals.

across time exhibits **temporal locality**, degrading smoothly with temporal separation. We further demonstrate that this locality is **directional**, with greater degradation from past training data to future evaluation data than in the reverse direction.

2.1 Pretraining Datasets are Non-Stationary

Pretraining corpora contain web data collected across many years, but these temporal slices are often treated as samples from a common underlying distribution. We first question this assumption: do the characteristics and composition of pretraining data remain stable over time?

We study the pretraining corpora FineWeb-Edu (Penedo et al., 2024b), DCLM-Baseline (Li et al., 2024), and Nemotron-CC v1[1] (Su et al., 2024), which span Common Crawl data from 2013–2025, 2013–2022, and 2013–2024, respectively. We use these datasets because they have varying filtering pipelines, which allows us to test whether temporal trends persist across distinctly curated corpora.

We begin by examining length and compressibility, which are simple, corpus-level properties that characterize document scale and how efficiently text can be represented. Related heuristics like document length and repetition are widely used for filtering in web-scale pretraining pipelines (Penedo et al., 2024b; Soldaini et al., 2024). Systematic temporal shifts in these quantities would suggest that crawl years are not temporally exchangeable in ways that are relevant to the construction of training corpora. Because years contain varying numbers of dumps with varying sizes, we aggregate dump-level metrics by year. In Figure 2, we find that median document length and GPT-2 tokenizer efficiency increase over time and zstd-3 compressed-size ratio decreases over time for all datasets, while Figure 6 shows the same trend for GPT-2 compressibility. Thus, while documents lengthen over time, multiple metrics show their text becomes more compressible. The consistency of these trends across datasets suggests that they are not tied to a particular filtering pipeline.

Next, we ask whether these surface-level changes are accompanied by changes in data composition. Such changes would indicate that models systematically encounter different types of content as the temporal range of their pretraining data changes. We measure shifts in part-of-speech (POS) and

[1]We do not use Nemotron v2 as it has no date-bearing field. Further, we restrict Nemotron to the three tiers in its organic partition (high, medium-high, medium), and we do not include synthetic or translated data.

domain distributions relative to the first year of each corpus in Figure 2, again averaging dump metrics into yearly statistics; for domains, we adopt the fastText domain classifier from OLMo-3 (Olmo et al., 2025). We find that the distance of both distributions from the 2013 baseline increases nearly monotonically over time, and Figure 6 shows the same trend in raw token distributions. Some domain shifts can be attributed to explicit events. For example, the sharpest yearly changes for all corpora occur at 2017, where the distinct host count doubles for FineWeb and triples for DCLM and Nemotron. This spike coincides with an expansion of Common Crawl's official coverage: in 2017, Common Crawl added URLs within two hops of the Alexa top-million domains (Nagel, 2017). Thus, temporal shifts can arise from changes in crawl coverage, not just changes in the web.

Finally, because POS and domain distributions capture predefined axes of corpus composition, we test whether temporal drift also appears in a broader semantic representation of the data. In Figure 2, we embed documents from each year with Qwen3-4B-Embedding (Zhang et al., 2025), and we plot L2 distance between the annual centroid and the 2013 baseline centroid over time. In Figure 6, we also plot this drift with Qwen3-0.6-Embedding and MiniLM-L6-v2 (Sentence Transformers, 2021). Across all encoder-dataset pairs, distance increases nearly monotonically over time. Due to the difficulty of interpreting centroid shifts in absolute terms, we compare these shifts to *inter-corpora* distances. We find that temporal drift within a corpus can rival distance between corpora, and it is largely driven by a shift in embedding location rather than local density (Section B).

We next consider possible sources of these temporal shifts. Sharp trends in later years are consistent with increased LLM-generated text. In Figure 6, we run the Desklib AI text detector (Desklib, 2025) and aggregate dump-level probabilities by year. For FineWeb and Nemotron, we find the estimated probability of AI-generated text is flat before the ChatGPT release at the end of November 2022 and rises sharply thereafter. The same trend appears in the frequency of four markers[2] associated with LLM-generated text; we do not analyze DCLM as its coverage does not extend into the LLM era. Because our other temporal trends begin well before 2022, increased LLM-generated text cannot fully explain our observed long-term shifts, but may contribute to their acceleration in recent years.

These trends are not clearly explained by filtering mechanisms. In Figure 6, FineWeb's annual mean educational score and DCLM's annual mean fastText score remain nearly constant across years, while Nemotron's quality composition only materially changes in the last two years. Thus, changes in the datasets' filtering signals do not appear sufficient to explain the observed temporal shifts.

Observation 1: Web pretraining data are non-stationary. Across three popular pretraining datasets, we find that lexical, information-theoretic, syntactic, topical, and semantic metrics shift systematically over time, with later years increasingly diverging from the initial distribution.

2.2 Generalization Across Time Degrades with Temporal Separation

While we have established substantial distributional shift in text data, it is not a priori clear that such shifts impact model performance. Thus, for each studied dataset, we independently pretrain a 411M model on each year at the Chinchilla-optimal token budget following DCLM conventions (Li et al., 2024), with precise details in Section C. Let $L(s, t)$ denote the year t test loss of a model pretrained on data from year s. Because absolute test loss can vary substantially across years, we report the *excess loss* $\Delta L(s, t) = L(s, t) - L(t, t)$ relative to a model pretrained on the test year's distribution, which normalizes for test year difficulty and focuses on the impact of temporal train-test mismatch.

Across all three corpora, excess loss $\Delta L(s, t)$ is smallest when pretraining and evaluation years are close, and grows smoothly with temporal separation $|t - s|$. Figure 3 shows this **temporal locality**: the Nemotron heatmap is lowest near the diagonal $t \approx s$, and aggregating model-evaluation pairs by signed year offset $t - s$ shows degradation in excess loss with increasing temporal separation across all three corpora. Thus, models generalize better between nearby years than between distant years.

Temporal effects are also **asymmetric**, with greater degradation when the training year is earlier than the evaluation year, which makes sense in light of the unidirectional nature of time. This effect is starkly shown in Figure 3 (right), where, for each pair of years, we compare transfer from the earlier year to the later year against the reverse direction. This asymmetry matters in practice because

[2]These markers are "delve", "as a large language model", "it's important to note", and "rich tapestry", originally defined and used by the FineWeb blog (Penedo et al., 2024a) in their temporal analysis.

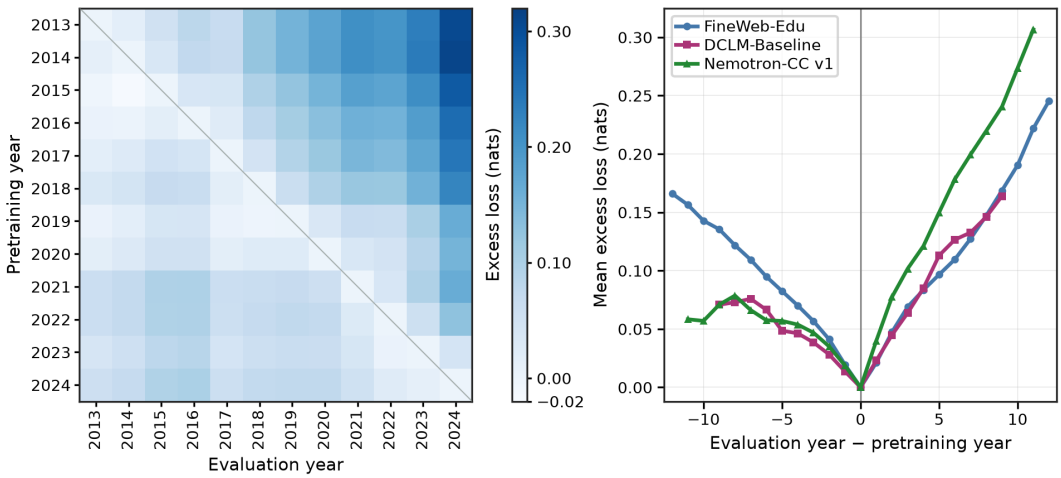

Figure 3: **Models generalize best between nearby years, with degradation increasing more rapidly when generalizing to future data.** *(Left)* Excess loss $\Delta L(s,t) = L(s,t) - L(t,t)$ for Nemotron-CC v1 is smallest near the diagonal and increases with temporal separation; FineWeb-Edu and DCLM-Baseline are in Section D. *(Right)* Averaging by year offset $t - s$ shows the same pattern across all corpora, with larger excess loss when generalizing to future data (positive offsets).

deployment is directional: future performance must be estimated using only data from the past, which is empirically the more difficult estimation problem. Together, non-stationarity and temporal locality suggest that the predictive value of evaluation data should depend on its age. In particular, recent evaluations may better estimate near-future model performance than older evaluations.

Observation 2: Model generalization across time is temporally local and directional. Models generalize better between nearby years than distant ones. Further, this degradation is asymmetric, as transfer from past training data to future evaluation data is worse than the reverse.

3 Time-Aware Evaluation Predicts Future Model Performance

In Section 2, we showed that pretraining corpora evolve over time and that models generalize better among nearby years than among more distant years. We now ask whether a similar temporal structure appears in model *evaluations*, and whether accounting for it allows us to better predict future model performance. As benchmark scores are used to inform model comparisons and deployment decisions, if older evaluations become less predictive of future performance, then pooling them with recent observations can distort performance claims and even reverse model rankings.

We study 12 datasets with temporal metadata, spanning biomedical QA, classification, summarization, citation prediction, and other scientific and professional domains (Table 1). Full dataset details are in Section E, and all models and inference settings are described in Section F.

We first show that recent evaluations better predict future model performance (Section 3.1) and then test whether this phenomenon persists at scale (Section 3.2). We then frame evaluation as a time-series forecasting problem and ask whether this can improve performance prediction (Section 3.3) and downstream model selection (Section 3.4). The trends we observe in predicting future model performance also persist in vision-language models, which we discuss in Section K.

3.1 Recent Evaluations Better Predict Future Performance

Benchmarks collapse performance across time into one score, even when the data span many years, which obscures the temporal structure observed in Section 2.2. We thus evaluate GPT-5.6 Sol/Luna (OpenAI, 2026) on StreamingQA (Liska et al., 2022) and SCIDOCA and group scores by year.

Table 1: We evaluate frontier and open-source models on 12 datasets of varying scale, task type, and temporal coverage. We construct all shaded datasets from the relevant source corpus (Section E).

Dataset	Task Description	Start Year	End Year	Examples
BioASQ	Biomedical QA	2014	2025	4,067
FinQA	Financial QA	1999	2019	8,281
StreamingQA	Evolving knowledge QA	2007	2020	16,307
SCIDOCA	Scientific citation prediction	2012	2025	70,000
NVD-VSP	CVSS severity scoring	2019	2025	116,711
NVD-RCM	CWE root-cause classification	2017	2025	132,103
CLERC	Legal citation matching	2006	2019	140,000
PubMed-MeSH	MeSH descriptor prediction	2010	2025	160,000
PubMed-Rhetorical	Rhetorical-role classification	2010	2025	160,000
PubMed-Summary	Constrained summarization	2010	2025	160,000
USPTO	CPC patent classification	2006	2024	190,000
WildTime-arXiv	arXiv category classification	2007	2022	2,057,952

StreamingQA is factual QA, where the temporal axis is the news article source date. SCIDOCA is scientific citation prediction, where the temporal axis is the publication date of the citing paper.

We find that performance changes substantially over time on both datasets, but in opposite directions (Figure 4, left panel). For both GPT-5.6 models, performance declines on SCIDOCA and improves on StreamingQA. Thus, even for a fixed model, a benchmark score depends meaningfully on the time period represented by the evaluation data.

Regardless of whether performance increases or decreases over time, recent performance better predicts future performance (Figure 4, right panel). We frame this as a forecasting problem. For each model and dataset, we use the score from year $t - \Delta$ to estimate performance in year t, and call this the **Δ-lag yearly predictor**. We see that the predictive error of the Δ-lag predictor increases monotonically with evaluation age Δ. Across all model-dataset pairs, the most recent evaluation is the best single-year predictor of next-year performance.

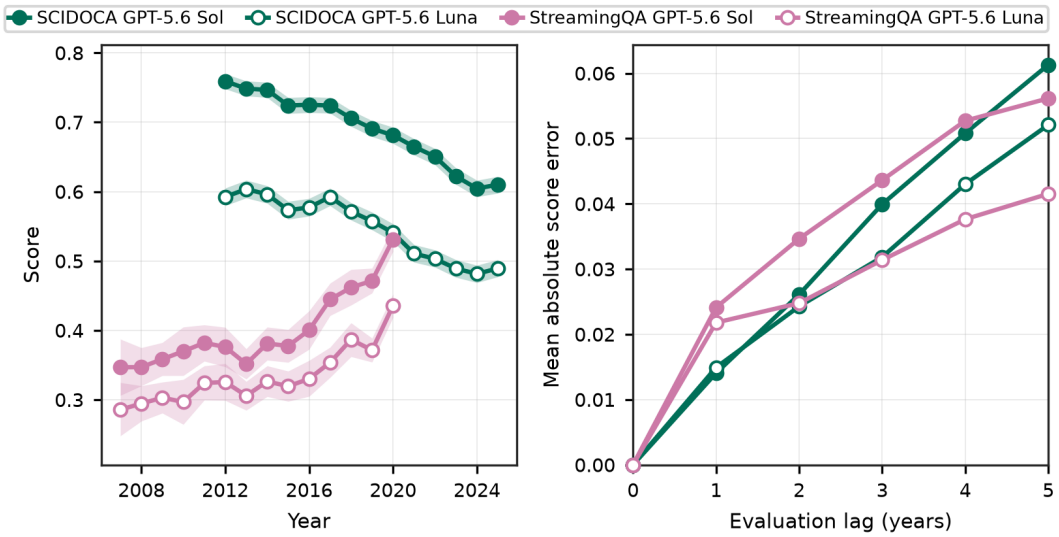

Figure 4: **Recent evaluations better predict future performance across different temporal performance trends.** We evaluate GPT-5.6 Sol/Luna on StreamingQA and SCIDOCA. *(Left)* Depending on the task and model, performance changes over time in distinct ways. *(Right)* Prediction error increases with evaluation age, so recent evaluations are the best predictors of future performance.

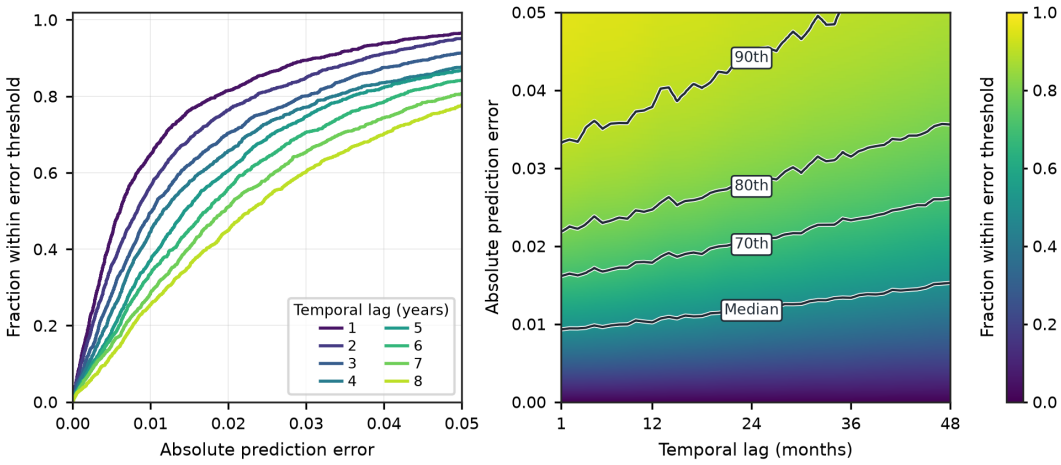

Figure 5: **Evaluation staleness smoothly degrades prediction quality at annual and monthly resolutions.** We evaluate 19 open-source models on 12 temporal datasets. *(Left)* Annual prediction error CDFs shift downward as lag increases from 1 to 8 years. *(Right)* Monthly prediction error rises as lag increases from 1 to 48 months. Error quantiles rise with lag, with the strongest degradation in the upper tail. Each comparison is restricted to periods where all plotted lags exist.

3.2 Temporal Locality Persists at Scale

The observations from the previous section show temporal locality on two tasks and a small set of frontier models, but they do not establish whether it persists more broadly across models and tasks. If stale evaluations systematically become less predictive across architectures, model scales, and task domains, then historical pooling is not only problematic for a few benchmarks. Rather, it would reflect a general misalignment between how benchmarks are scored and how their scores are used.

We test this at scale using 19 open-source models from the Qwen3 (Yang et al., 2025), OLMo 2 (OLMo et al., 2024), Llama 3.1 (Grattafiori et al., 2024), and Gemma 3 (Team et al., 2025) families on 12 temporally structured datasets, evaluating yearly and monthly[3] performance.

At yearly resolution, increasing the evaluation staleness Δ monotonically shifts the error distribution of Δ-lag yearly predictors towards larger errors across models and datasets (Figure 5, left). This ordering holds across error thresholds, so the advantage of recent evaluations is robust across varying error tolerances. At monthly resolution, the predictive accuracy of **Δ-lag monthly predictors** also degrades smoothly with age Δ, rather than changing suddenly at a particular staleness threshold (Figure 5, right). We corroborate this pattern with a Wasserstein analysis in Section G. Further, the quantile curves show that degradation is greater in the upper tail, so stale evaluations disproportionately increase large forecasting errors.

3.3 Forecasting Model Performance

Because benchmark scores are used to inform expectations about future model performance, benchmark evaluation is already implicitly a forecasting problem. For each target year or month, forecasters receive only evaluations from earlier periods and predict performance in the subsequent period. We represent standard pooled benchmark scoring with a historical mean forecaster that ignores when evaluations were collected. Temporal locality suggests that discarding such temporal structure can substantially degrade predictive accuracy.

We now compare the historical mean baseline with **persistence**, which predicts performance using only the most recent observation, equivalent to our 1-lag yearly or monthly predictor. We then test whether explicitly modeling the trajectory improves prediction[4], using the Chronos-2 time-series foundation model (Ansari et al., 2025) and four classical methods: linear extrapolation, ARIMA

[3]We exclude FinQA and BioASQ from monthly analysis due to poor date recovery or very sparse coverage.

[4]We restrict our analysis to target periods with a contiguous history of at least 5 years or 12 months.

selected by AICc (Hurvich & Tsai, 1989), Simple Exponential Smoothing (SES) (Holt, 1957), and damped Holt (Gardner Jr & McKenzie, 1985). In the left panel of Figure 1, we plot error CDFs of selected annual forecasters; all annual and monthly forecasters and their MAEs are in Section H.

At annual resolution, persistence reduces MAE by roughly one-third relative to the historical mean. None of the classical time-series methods outperform persistence, while Chronos-2 does so by up to several points; in other words, the most recent observation already captures most available annual predictive signal. At monthly resolution, however, the picture changes substantially. Persistence reduces MAE by 19.8% relative to the historical mean, and *every explicit time-series forecaster improves further upon persistence*. Chronos-2, in particular, is remarkably effective at all scales.

Observation 3: Time-aware evaluation better predicts future model performance. Time-aware forecasters substantially outperform the historical mean, which represents temporal pooling in standard benchmark evaluation. At annual resolution, most of the benefit comes from the most recent observation, while explicitly modeling the trajectory offers further gains for denser monthly histories. Thus, when evaluations are used to anticipate future model performance, we recommend that they retain their temporal ordering rather than being aggregated into one score.

3.4 Ranking Models Under Benchmark Staleness

Above, we found that time-series forecasters can substantially improve predictions of future model performance. These forecasts support various deployment choices, such as determining if a model is likely to clear a task-specific performance requirement, or choosing among candidate models. Pooling evaluations across time is problematic because it can obscure changes in model rankings. Section I shows ranking inversions between thinking and non-thinking variants, different model sizes in a family, similarly capable models across families, and frontier versus open-weight models.

We now ask whether time-series forecasters improve modeling and deployment choices. We focus on model selection: predicting which model from a candidate set will perform best on new data. We give a forecaster K models and select the model it predicts will achieve the highest performance on new data. We measure how often this procedure selects the model that actually performs best in the target period, averaged over all subsets of size K drawn from 19 total models and pooled across benchmarks and target periods. We call the resulting measurement the **optimal selection rate**.

The optimal selection rate for any method naturally degrades as the shortlist grows, but time series forecasters perform significantly better than historical pooling (Figure 1, right panel). We provide a concrete failure case of historical pooling on WildTime-arXiv (Yao et al., 2022) in Section I.

Perhaps surprisingly, at monthly resolution, only Chronos-2 improves model selection over the historical mean; persistence performs worse despite reducing forecasting MAE (Figure 11, left panel). We analyze this discrepancy in Section J, where we show that much of persistence's forecasting gains come from errors shared across models, which do not affect their ranking. Thus, lower forecasting error does not necessarily translate into better model selection.

Optimal selection rate treats all mistakes equally, even though selecting a near-tied model is less consequential than selecting one that performs far worse. This motivates us to consider selection **regret**, the accuracy gap between the selected and optimal models. We define high-regret choices as those exceeding two accuracy points. At annual resolution, persistence and Chronos-2 roughly halve the rate of high-regret choices relative to the historical mean baseline (Figure 11, right panel). As in optimal model selection, only Chronos-2 reduces this rate at monthly resolution. Thus, time-aware evaluation can reduce not only forecast error but also the frequency of high-regret model selections.

Observation 4: Time-aware evaluation improves model selection. Using temporally pooled scores to select a model from a shortlist for deployment can preserve stale model rankings and cause highly suboptimal selections, especially as the shortlist grows. Recent evaluations provide a stronger basis for annual model selection, while denser monthly histories benefit from explicitly modeling the trajectory. Model selection should account for when evaluation data were collected.

4 Temporal Recency Impacts Training Data Selection

We have shown that temporal shifts affect model generalization and predictions of future performance. We now ask whether these shifts should inform how practitioners select training data. Pretraining pipelines already optimize data via deduplication, quality filtering, source and domain balancing, and stage-specific mixtures that trade off diversity and quality (Olmo et al., 2025; Blakeman et al., 2025). We argue that *recency* should also be considered in data filtering and sampling rates.

We first ask whether restricting pretraining to recent data improves near-future performance. Extending the experiments in Section 2.2, at each cutoff we pretrain on all FineWeb data up to that cutoff (cumulative) or just the last five years (recent-window), and compare next-year perplexity on FineWeb, PubMed, and arXiv. We repeat each comparison on five training recipes[5].

On FineWeb and PubMed, recent-window usually has lower next-year perplexity than cumulative, whereas arXiv shows no clear advantage from pretraining on recent data (Section L). Prior work has similarly shown that replaying older data can improve retention, and the optimal balance between new and old data is domain-dependent (Li et al., 2025). Thus, we do not suggest discarding old data. Practitioners could instead decrease sampling rates with age while retaining historical coverage.

Because the appropriate level of recency depends on the target domain, we use the most recent yearly evaluation to choose between recent-window and cumulative. This procedure selects the better next-year model in 97% of cases across all datasets. Sensitivity to evaluation age is domain-dependent: on FineWeb, selection accuracy falls to 51% with five-year-old evaluations, while PubMed and arXiv selection accuracy remain stable across lags (Section L). For data mixture search methods such as DCLM and NVIDIA CLIMB (Li et al., 2024; Diao et al., 2026), this suggests adding recency to the search space and using target-relevant, time-forward validation to select among candidate mixtures.

We next ask whether the same principle extends to late-stage training, where practitioners deliberately reshape the data mixture. OLMo 3 curates high-quality math, science, code, and instruction-following data (Olmo et al., 2025), Qwen3 increases STEM, coding, and reasoning data (Yang et al., 2025), and CPT recipes optimize the composition of adaptation data (Parmar et al., 2024).

For each pretrained checkpoint[6], we perform CPT on 1.6M tokens of PubMed or arXiv data from each year of the pretraining window. In both domains, five-year-old CPT data perform worse than current-year CPT data at every cutoff, making the most recent adaptation data a strong CPT baseline.

However, adaptation recency is not always optimal in CPT. For PubMed, the most recent CPT year always minimizes next-year perplexity. By contrast, it does so for only 12 of 18 windows on arXiv. Selecting the candidate that performs best on the current year improves this to 14 of 18 and reduces mean next-year selection regret by 87%. Thus, practitioners should preserve timestamps and treat adaptation year as an explicit data selection variable, using the most recent data when performance is consistently ordered by adaptation age, and recent target-domain evaluation otherwise.

Observation 5: Temporal recency matters for training data selection. The most recent data are often, but not always, preferable for pretraining and continued pretraining. When the best temporal choice depends on the target domain, target-relevant validation can identify which training-data policy generalizes best to the near future.

5 Discussion

Our results suggest that LLM evaluation should be treated as a forecasting problem. Recent evaluations predict future performance better than standard pooled benchmarks, and explicit time-series forecasters further improve prediction and model selection. In deployment, accounting for temporal structure reduces prediction errors and high-regret model choices. This perspective also informs training data selection, where recency can improve near-future performance but is not always optimal. More broadly, we recommend that practitioners preserve temporal information so that evaluation and data selection reflect the evolution of data distributions and model performance over time.

[5] We use the five learning-rate and weight-decay configurations from 400M-parameter tuning in DCLM.

[6] We select the best training recipe for each cumulative and recent-window model. The best recipes are almost constant across evaluation years, suggesting that temporal sensitivity is not due to poor optimization.

AI USE STATEMENT

In this work, we used generative AI tools to polish writing and figures, search for related work, and search for source corpora with sufficient yearly coverage for our temporal analysis. We have not used generative AI tools to perform any research ideation or execution, to draft sections of the paper, or to generate synthetic datasets; proving mathematical claims is not applicable to this work. We have reviewed all AI-assisted work. For example, we checked all LLM-generated corpora suggestions through a manual literature survey, and we carefully verified relevant data processing techniques from prior work and cited any such works in Section E. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We include full details to reproduce our dataset constructions from source corpora in Section E, as well as full details on dataset scoring and model inference in Section F. We further include full details for our yearly and multi-year pretraining experiments in Section C.

REFERENCES

Md Tanvirul Alam, Dipkamal Bhusal, Le Nguyen, and Nidhi Rastogi. Ctibench: A benchmark for evaluating llms in cyber threat intelligence. *Advances in Neural Information Processing Systems*, 37:50805–50825, 2024.

Md Tanvirul Alam, Dipkamal Bhusal, Salman Ahmad, Nidhi Rastogi, and Peter Worth. Athenabench: A dynamic benchmark for evaluating llms in cyber threat intelligence. In *2025 Annual Computer Security Applications Conference Workshops (ACSAC Workshops)*, pp. 443–450. IEEE, 2025.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

David Bailey, Jonathan Borwein, Marcos Lopez de Prado, and Qiji Jim Zhu. The probability of backtest overfitting. *The Journal of Computational Finance*, 20(4):39–69, 2017.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, et al. Nvidia nemotron 3: Efficient and open intelligence. *arXiv preprint arXiv:2512.20856*, 2025.

Cody Blakeney, Mansheej Paul, Brett W Larsen, Sean Owen, and Jonathan Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training. *arXiv preprint arXiv:2406.03476*, 2024.

Harold Booth, Doug Rike, and Gregory A Witte. The national vulnerability database (nvd): Overview. 2013.

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.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, 2021.

Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. Dated data: Tracing knowledge cutoffs in large language models. *arXiv preprint arXiv:2403.12958*, 2024.

Cornell University. arXiv Dataset. Kaggle dataset identifier: Cornell-University/arxiv; accessed September 2026.

Jack Cushman, Matthew Dahl, and Michael Lissner. eyecite: A tool for parsing legal citations. *Journal of Open Source Software*, 6(66):3617, 2021.

An Dao, Vu Tran, Le-Minh Nguyen, and Yuji Matsumoto. Overview of scidoca 2025 shared task on citation prediction, discovery, and placement. *arXiv preprint arXiv:2509.24283*, 2025.

Franck Dernoncourt and Ji-Young Lee. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 308–313, 2017.

Desklib. Ai text detector v1.01. Hugging Face, 2025. URL https://huggingface.co/desklib/ai-text-detector-v1.01.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, et al. Nemotron-climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. *Advances in Neural Information Processing Systems*, 38, 2026.

Lorenzo Emer, Marco Lippi, Andrea Mina, and Andrea Vandin. Large language models for patent classification: Strengths, trade-offs, and the long tail effect. *arXiv preprint arXiv:2601.23200*, 2026.

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

Everette S Gardner Jr and ED McKenzie. Forecasting trends in time series. *Management science*, 31(10):1237–1246, 1985.

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

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8342–8360, 2020.

Elise Han, Chengpiao Huang, and Kaizheng Wang. Model assessment and selection under temporal distribution shift. *arXiv preprint arXiv:2402.08672*, 2024.

Harvard Law School Library Innovation Lab. Caselaw access project, 2018. URL https://case.law/. Harvard University.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *ONR Research Memorandum No. 52*, 1957.

Abe Bohan Hou, Orion Weller, Guanghui Qin, Eugene Yang, Dawn Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. Clerc: A dataset for us legal case retrieval and retrieval-augmented analysis generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7913–7928, 2025.

Clifford M Hurvich and Chih-Ling Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6237–6250, 2022.

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, et al. The semantic scholar open data platform. *arXiv preprint arXiv:2301.10140*, 2023.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific data*, 10 (1):170, 2023.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363, 2021.

Yann LeCun. Learning algorithms for classification: A comparison on handwritten digit recognition. 1995.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*, 2024.

Jeffrey Li, Mohammadreza Armandpour, Seyed Iman Mirzadeh, Sachin Mehta, Vaishaal Shankar, Raviteja Vemulapalli, Samy Bengio, Oncel Tuzel, Mehrdad Farajtabar, Hadi Pouransari, et al. Tic-lm: A web-scale benchmark for time-continual llm pretraining. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32231–32273, 2025.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 2356–2362, 2021.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien De Masson D'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pp. 13604–13622. PMLR, 2022.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4969–4983, 2020.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60, 2014.

Sebastian Nagel. January 2017 crawl archive now available. Common Crawl Blog, February 2017. URL https://commoncrawl.org/blog/january-2017-crawl-archive-now-available.

Kai Nakamura, Sharon Levy, and William Yang Wang. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. *arXiv preprint arXiv:1911.03854*, 2019.

National Library of Medicine. PubMed 2026 Baseline Released. NLM Technical Bulletin, No. 468, e6, 2026. URL https://www.nlm.nih.gov/pubs/techbull/jf26/jf26_PubMed_2026_BaselineRelease.html. Posted January 30, 2026.

Kai Nylund, Suchin Gururangan, and Noah A Smith. Time is encoded in the weights of finetuned language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2571–2587, 2024.

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

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

OpenAI. GPT-5.6 System Card, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

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

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Colin Raffel, Leandro Werra, and Thomas Wolf. FineWeb: decanting the web for the finest text data at scale, May 2024a. URL https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin 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, 2024b.

Hippolyte Pilchen, Romain Fabre, Franck Signe Talla, Patrick Perez, and Edouard Grave. Understanding data temporality impact on large language models pre-training. *arXiv preprint arXiv:2605.22769*, 2026.

Subhash Chandra Pujari. *Neural Patent Classification Beyond Title and Abstract: Leveraging Patent Text and Metadata*. PhD thesis, 2024.

Subhash Chandra Pujari, Annemarie Friedrich, and Jannik Strötgen. A multi-task approach to neural multi-label hierarchical patent classification using transformers. In *European conference on information retrieval*, pp. 513–528. Springer, 2021.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Anna M. Ripple, Lou S. Knecht, and Frances Truong. Structured abstracts in MEDLINE: Newly identified and mapped labels available. *NLM Technical Bulletin*, (407):e1, 2015. URL https://www.nlm.nih.gov/pubs/techbull/nd15/nd15_structured_abstracts.html. November–December 2015.

Manley Roberts, Himanshu Thakur, Christine Herlihy, Colin White, and Samuel Dooley. Data contamination through the lens of time. *arXiv preprint arXiv:2310.10628*, 2023.

Sentence Transformers. all-minilm-l6-v2. Hugging Face, 2021. URL https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. 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)*, pp. 15725–15788, 2024.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. *arXiv preprint arXiv:2412.02595*, 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 preprint arXiv:2503.19786*, 2025.

United States Patent and Trademark Office. Final release of patentsview metadata, pre-grant and granted (12/31/2024), 2025.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. Freshllms: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, 2024.

David Wadden, Kejian Shi, Jacob Morrison, Alan Li, Aakanksha Naik, Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, Luca Soldaini, et al. Sciriff: A resource to enhance language model instruction-following over scientific literature. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6083–6120, 2025.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, et al. Livebench: A challenging, contamination-limited llm benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 91595–91631, 2025.

Halbert White. A reality check for data snooping. *Econometrica*, 68(5):1097–1126, 2000.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pp. 1253–1256, 2017.

Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. *Advances in Neural Information Processing Systems*, 35:10309–10324, 2022.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Zeyu Zhang and Bradly C Stadie. Temporal leakage in llm backtesting: Measurement, validation, and adjusted scores. *arXiv preprint arXiv:2608.02985*, 2026.

A Related Work

Temporal distribution shift and adaptation. Prior work has shown that text distributions and model performance shift over time. Lazaridou et al. (2021) show that language models degrade on text beyond their cutoff, while Yao et al. (2022) benchmark natural temporal distribution shifts across many domains. Cheng et al. (2024) further show that a model's effective knowledge cutoff wildly varies across data sources. Other works explicitly adapt models over time: TemporalWiki uses continual updates on evolving knowledge (Jang et al., 2022), TiC-LM uses continual pretraining on 114 Common Crawl dumps (Li et al., 2025), Nylund et al. (2024) derive temporal model updates that generalize to the future, and Pilchen et al. (2026) find that chronological ordering improves factual freshness in pretraining. Our work instead provides a unified framework connecting temporal pretraining data shifts, evaluation staleness, forecasting, model selection, and training data selection.

Time-aware evaluation and model selection. Some benchmarks incorporate time through timestamped or continually refreshed evaluation data (Liska et al., 2022; Vu et al., 2024; White et al., 2025; Jain et al., 2025). Temporal information has also been used as a signal of benchmark contamination. Roberts et al. (2023) compare model performance on problems before and after training cutoffs, while Zhang & Stadie (2026) show that post-cutoff performance decay depends on benchmark construction. We use temporal information to study a different question: how does the age of an evaluation impact its ability to predict future performance? Most closely related to our work, Han et al. (2024) studies model assessment and selection under temporal distribution shift, using adaptive rolling windows to estimate *current* generalization error from current and historical data. We instead treat benchmark evaluation as a forecasting problem, using temporally ordered evaluations to predict performance in the *future*, and studying how model selection changes as evaluations age.

Training data selection and continued pretraining. Prior work has studied how training data composition impacts model performance. Domain-adaptive pretraining improves performance on a target domain (Gururangan et al., 2020), DSIR selects pretraining data to match a target distribution (Xie et al., 2023). Later work on late-stage training studies domain upsampling (Blakeney et al., 2024) and data selection for continued pretraining (Parmar et al., 2024). Time-continual approaches further study how models should incorporate new data while retaining older knowledge (Li et al., 2025). We complement these approaches by treating *data recency* as an explicit design variable and using recent, time-forward validation to select training data policies for near-future performance.

B Additional Temporal Shifts in Pretraining Corpora

Figure 6 shows additional diagnostics supporting the temporal distribution shifts in pretraining corpora. As embedding-centroid distances are difficult to interpret in absolute terms, we compare them with distances between corpora. Temporal drift within a single corpus reaches roughly the full 2013 DCLM-Nemotron distance and half the 2013 FineWeb-DCLM distance, showing that within-corpus temporal variation can rival the differences between distinct corpora. This drift largely reflects a shift in embedding location rather than local density: for both Qwen encoders, changes in mean 10th-nearest-neighbor distance over time are less than 10% of the corresponding centroid shift. The figure also reports additional document, token, AI-text, and dataset-specific quality metrics.

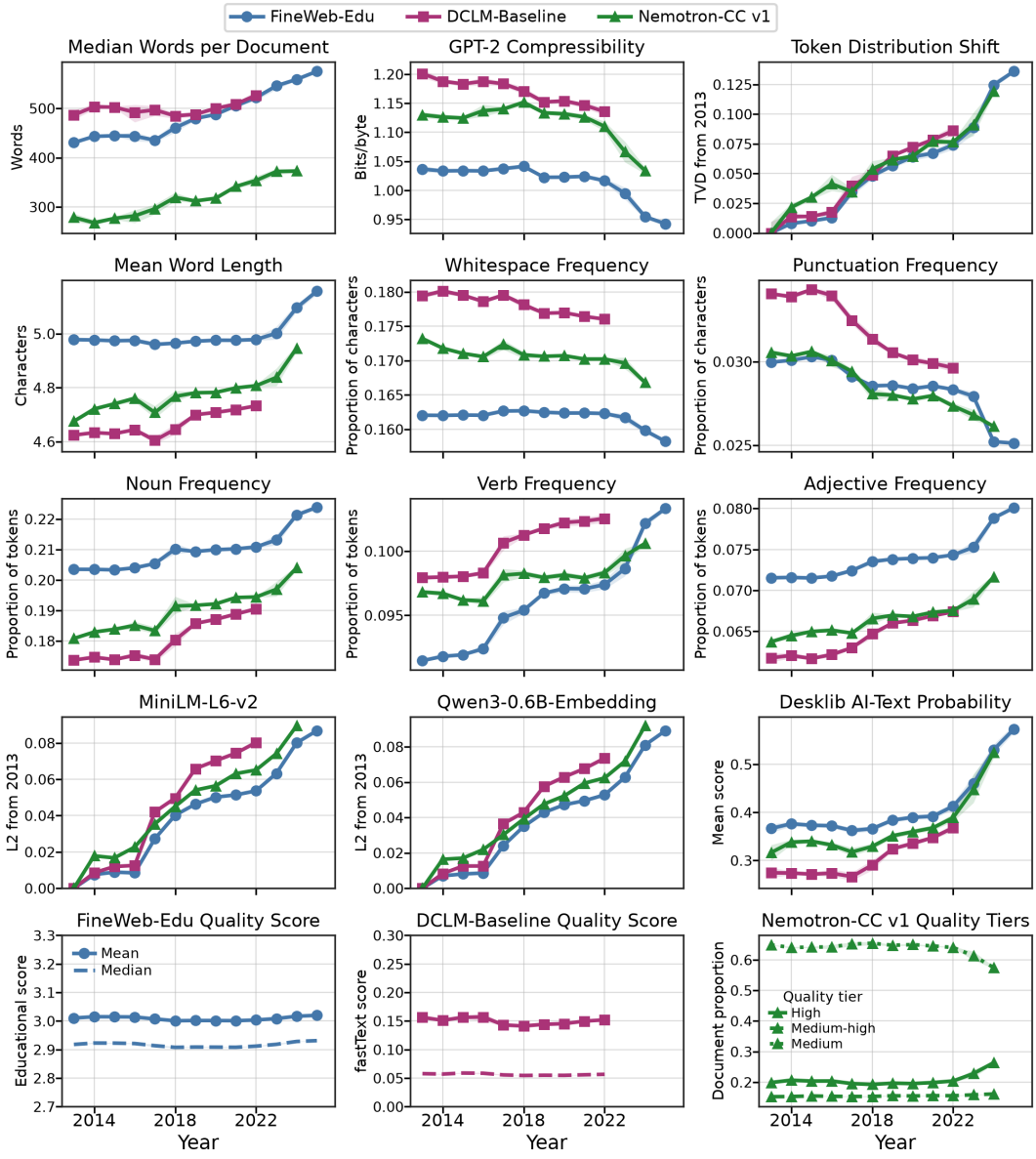

Figure 6: **Additional corpus metrics show systematic temporal distribution shift across pretraining corpora.** FineWeb-Edu, DCLM-Baseline, and Nemotron-CC v1 exhibit changes in document statistics and token and part-of-speech composition *(first three rows)*. Annual document-embedding centroids shift over time, while estimated AI-text probability rises sharply in recent years *(fourth row)*. Dataset-specific quality signals remain relatively stable *(fifth row)*.

C TRAINING DETAILS

We follow the standard DCLM architecture (Li et al., 2024). Each model has 411,665,408 trainable parameters with 24 pre-norm transformer layers. Each layer has width 1024, contains 8 attention heads of dimension 128, and contains a 2,816-dimensional bias-free MLP with SwiGLU. As in DCLM, we use RoPE, QK-normalization, LayerNorm, untied input and output embeddings, and no dropout. The tokenizer is GPT-NeoX. Each model is trained on exactly 4,019,200 packed sequences of length 2,048 (a total of 8,231,321,600 tokens), using 7,850 optimizer steps with a global batch size of 512 sequences. Documents are separated by EOS and packed without padding, following the standard DCLM convention. Evaluation is carried out on 32,768 fixed-length sequences, for a total of 67,108,864 tokens. In training, we use AdamW with $\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-8}$. We use a linear warmup for the first 2000 steps to the peak learning rate, then cosine decay to 3×10^{-5}. The gradient norm is clipped to 1, and we include a 10^{-4} z-loss term. We conduct training on NVIDIA B200s, and we use BF16 autocast and seed 124. Continued pretraining uses the exact same settings, with the warmup step count proportionally scaled to the token count.

D ALL EXCESS LOSS HEATMAPS

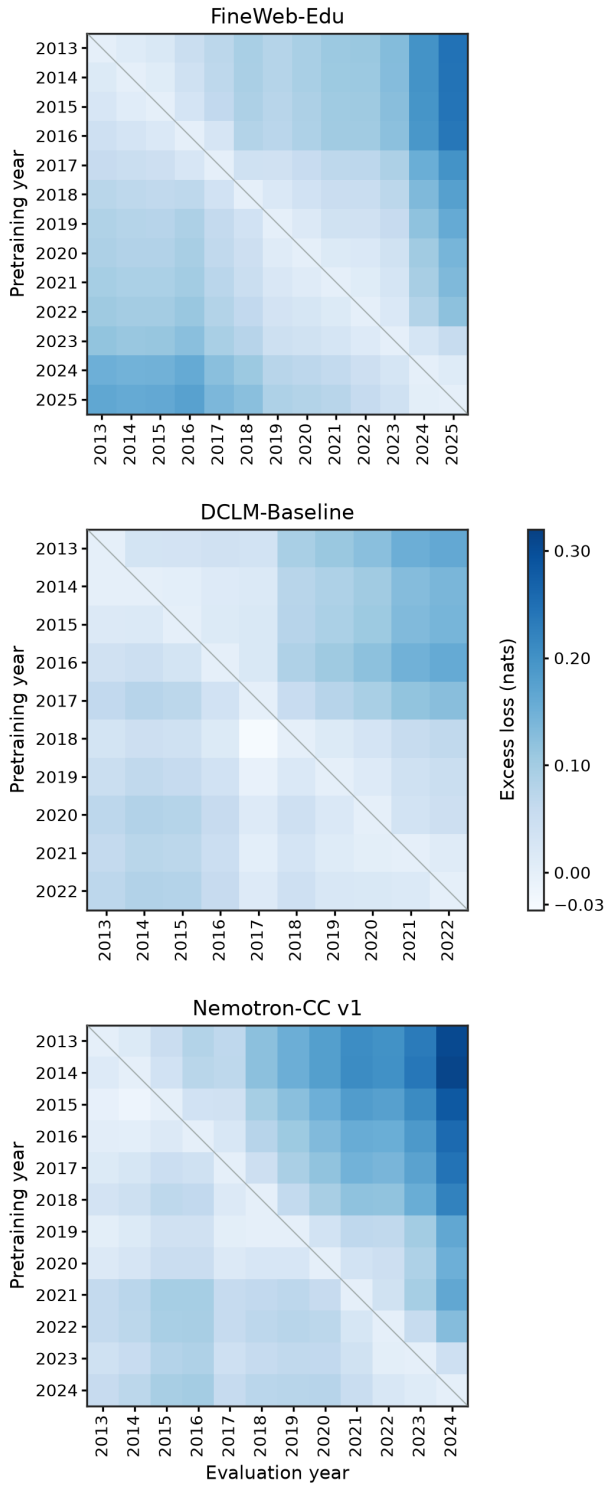

Figure 7: **Models generalize best to nearby years across all three pretraining corpora.** On FineWeb-Edu, DCLM-Baseline, and Nemotron-CC v1, excess loss $\Delta L(s,t) = L(s,t) - L(t,t)$ is smallest near the diagonal and increases as the pretraining year s and evaluation year t separate.

E DATASETS

E.1 BIOASQ TASK B

We construct a temporal biomedical QA benchmark from BioASQ Task B (Krithara et al., 2023), using the official Golden Enriched test sets from BioASQ 2B through 13B (2014–2025). Phase B provides gold supporting snippets, so our task is context-grounded QA rather than closed-book QA. We process each annual set independently using the SciRIFF BioASQ preprocessing and instruction templates (Wadden et al., 2025), preserving the original annual splits instead of using SciRIFF's repartitioning. We use template 0 with its 4,096-token configuration and drop overlength examples. SciRIFF concatenates gold snippets into the context, maps factoid synonym lists to the longest answer string, and keeps the first synonym for each entity in list questions. We exclude summary questions because they require judge-based evaluation. Of the 5,486 raw questions, we retain 4,067 yes/no, factoid, and list questions with gold answers.

Models receive the question with concatenated gold snippets and must generate an answer in the format required by the question type. Yes/no responses are scored by exact accuracy after stripping whitespace and lowercasing. Factoid responses use exact match against the single SciRIFF-converted reference after lowercasing, removing punctuation and articles, and collapsing whitespace. List responses are parsed with SciRIFF-style JSON extraction, normalized with SciRIFF's list normalization and compared as sets using exact-entry F1 scores, with unparseable responses receiving a score of zero. The annual score is the mean per-question score across all examples.

The temporal axis is the challenge year. We do not perform monthly analysis because each year's test questions are concentrated in a short challenge window, making monthly evaluation highly sparse.

E.2 STREAMINGQA

We construct a temporal evolving-knowledge QA benchmark from StreamingQA (Liska et al., 2022), which is built from English WMT News Crawl articles. We start from the official 36,378-example evaluation file and retain only examples with `recent_or_past = past`, as these span the historical corpus rather than clustering at 2020. This gives 16,307 examples from August 2007 to December 2020. We preserve the original question, its three official reference answers, the question timestamp, and the source-article timestamp.

Models are evaluated with closed-book QA. Following StreamingQA, each question is prefixed with "Today is {date}" using the question timestamp. We append a fixed instruction requesting only the short answer. Predictions are scored with SQuAD-style token F1 (Rajpurkar et al., 2016): prediction and reference strings are lowercased, ASCII punctuation and articles are removed, and whitespace is collapsed. We compute token F1 against each of the three official references and keep the maximum. The cohort score is the mean per-question F1.

The temporal axis is the source-article timestamp, not the question timestamp. Monthly analysis uses the source-article calendar month directly, requiring no additional metadata or matching. The full span contains 161 months, of which 151 have data and 10 are represented as missing observations.

E.3 FINQA

We construct a temporal financial numerical-reasoning benchmark from FinQA (Chen et al., 2021). We combine train, development, and test splits to get 8,281 examples. FinQA uses S&P 500 earnings reports from 1999–2019. We keep all examples and assign report years from the canonical ID.

For each example, the model receives the complete `pre_text`, the full table rendered in Markdown, the complete `post_text`, and the question, all in source order. The Markdown conversion preserves table structure while also handling irregular row lengths and cell formatting. Gold supporting facts, programs, execution results, reasoning steps, and answers are not included in the input.

FinQA scoring is more involved because of heterogeneous answer formatting. For parseability, we request a final answer in the form "Final answer: ...". For scoring, `qa.answer` is used if it is yes/no or contains exactly one percentage quantity; otherwise it must contain exactly one number. If unusable, the same rule is applied to `qa.steps[-1].res`. Predictions are extracted from the final

answer marker, then conclusion phrases, and then last-line fallbacks. Parsing supports signed and comma-separated numbers, currency, percentages, leading decimals, accounting parentheses, and ratios and fractions, but never computes unfinished arithmetic. Numeric predictions are compared after rounding both prediction and reference to the number of decimal places shown in the reference, and percentage and non-percentage values must also match in type. Ambiguous outputs score zero.

We do not perform monthly analysis on FinQA because the processed benchmark only gives report years and it is difficult to externally reconstruct all filing dates.

E.4 WILDTIME-ARXIV

We use the arXiv dataset from Wild-Time (Yao et al., 2022), a 172-way classification task that predicts an arXiv paper's primary subject category from its title. It includes 2,057,952 examples from 2007–2022, with the temporal axis being the arXiv metadata field `update_date` rather than initial submission year. Models receive the paper title and the fixed list of 172 valid category codes, and must return exactly one code. We score 172-way classification accuracy: a response is correct if it contains exactly one valid category from the Wild-Time label set and matches the gold label.

We download the official `arxiv.pkl` from the Wild-Time repository, which retains lowercased titles and integer class IDs but not category strings. We recover the 172 category strings by matching titles against Cornell's `arxiv-metadata-oai-snapshot.json` (Cornell University). We verify that the recovered mapping reproduces Wild-Time's class ID ordering. Because the arXiv taxonomy changes over time, we preserve Wild-Time's original label space instead of canonicalizing labels. We restore three historical labels from the original Wild-Time label space, replacing current labels `math.NA`, `eess.SY`, and `econ.GN` with `cs.NA`, `cs.SY`, and `q-fin.EC`.

For monthly analysis, we recover month-level dates from the same metadata. Each Wild-Time example is matched by its title to a current arXiv record, and we accept its `update_date` only if its year agrees with the official Wild-Time year. We discard examples without a unique valid match. Yearly analysis retains all examples, while monthly data retain 94.3% of original examples.

E.5 PUBMED

We construct PubMed-Summary, PubMed-MeSH, and PubMed-Rhetorical from the 2026 PubMed Baseline (39,994,988 records) released by NLM (National Library of Medicine, 2026). We assign each record to the earliest year available in `PubDate` or `ArticleDate` and retain 2010–2025. Summary and MeSH share the same 160,000 English MEDLINE records with nonempty abstracts and at least one MeSH descriptor, sampled at 10,000 random records per year. Rhetorical independently samples 10,000 English MEDLINE randomized controlled trials per year, requiring 3–9 structured-abstract sections whose publisher labels map to the five PubMed-200k-RCT roles (Dernoncourt & Lee, 2017; Ripple et al., 2015).

Summary provides the title and abstract and requests exactly three bullet lines totaling at most 50 words. Each example receives a binary instruction-following score for satisfying both constraints. MeSH provides the title and abstract and requests a JSON string array of MeSH descriptors. Predictions are mapped to descriptor IDs using the 2026 MeSH vocabulary, and each article is scored with set F1. Unresolved strings are false positives and invalid arrays are empty predictions. Rhetorical sentence-segments each structured abstract with Stanford CoreNLP (Manning et al., 2014), assigns each sentence its section role, and removes section headings that leak the label, giving 1,978,916 sentences. Models get numbered sentences and predict one of five rhetorical roles for each, and performance is mean per-abstract sentence accuracy.

For monthly analysis, we rejoin sampled PMIDs to the same baseline XML. We assign a month only when the publication month is unambiguous and consistent with the annual cohort. This retains 142,553 of 160,000 Summary/MeSH and 146,950 of 160,000 Rhetorical examples in 192 months.

E.6 USPTO

We construct a temporal patent-classification benchmark similar to USPTO-70k (Pujari et al., 2021). The task is to predict four-character Cooperative Patent Classification (CPC) subclasses from U.S. patent titles and abstracts. We use the final 2024 PatentsView metadata release (United States Patent

and Trademark Office, 2025) and follow the construction from Pujari (2024), aggregating current CPC assignments by patent and keeping all CPC subclasses. For each grant year from 2006–2024, we keep utility patents with non-empty titles and abstracts and at least one CPC subclass. This yields 5,152,802 eligible patents, and we sample 10,000 patents per year to obtain 190,000 total examples.

Models receive the patent title and abstract and output up to seven four-character CPC subclasses in JSON format, like in Emer et al. (2026). We normalize predicted labels to uppercase four-character codes, deduplicate them, and treat unparseable outputs as empty. Each example is scored with set F1 against the full set of gold CPC subclasses, and the cohort score is the mean across examples.

The temporal axis is the patent grant date from PatentsView. Monthly analysis uses the grant month.

E.7 NVD

We construct temporal versions of CTIBench's Root-Cause Mapping (RCM) and Vulnerability-Severity Prediction (VSP) tasks (Alam et al., 2024). Because NVD yearly feeds are partitioned by CVE-ID prefix and not publication date, we pool all feeds before forming temporal cohorts. RCM predicts a CWE identifier from a CVE description, while VSP predicts its CVSS v3.1 base vector. We use AthenaBench's NVD filtering and prompt templates (Alam et al., 2025). We use the official NVD JSON 2.0 feeds for 2002–2025 (Booth et al., 2013) and apply AthenaBench-style filtering. For RCM, we keep English descriptions with exactly one valid CWE label and at least 25 words. For VSP, we require a CVSS v3.1 record, prefer NVD-authored metrics over CNA metrics when available, and require at least 30 words. We globally deduplicate descriptions, retaining 132,103 RCM examples from 2017–2025 and 116,711 VSP examples from 2019–2025.

We use the official AthenaBench prompt templates. RCM requests a short justification followed by a CWE identifier and is scored with exact CWE accuracy. VSP requests the eight CVSS base metrics and a final CVSS vector. An example is correct if all eight predicted components match the gold vector, with incomplete predictions scored as zero. This exact-vector score differs from AthenaBench, which evaluates VSP using CVSS-score deviation.

The temporal axis is the NVD `published` timestamp, not the CVE-ID year or `lastModified`. Monthly analysis uses the publication month directly, requiring no external metadata.

E.8 SCIDOCA

We construct SCIDOCA from SCIDOCA 2025 Task 2 (Dao et al., 2025), using the frozen Semantic Scholar 2026-09-01 S2ORC and metadata release (Lo et al., 2020; Kinney et al., 2023). Source papers must contain body text and citation annotations that can be linked to cited papers. We assign each citation to the paragraph that contains it, and keep only paragraphs where all citations are resolved and at least one non-cited candidate paper is available as a distractor. Citations are replaced by `[CITATION_i]` masks. Following SCIDOCA, the candidate set for each example includes all distinct papers cited anywhere in the source paper.

We join candidate titles and abstracts from Semantic Scholar metadata. Candidates lacking either are removed, as are citation masks whose gold papers lack complete metadata. Examples must retain at least one gold citation and one distractor. Within each year, we rank eligible source papers, consider up to the first 15,000 before metadata filtering, and retain the first 5,000 surviving examples. This produces 70,000 examples from distinct source papers spanning 2012–2025.

Models receive the masked paragraph and each candidate's CorpusId, title, and abstract, and return a JSON `predicted_citations` mapping. Predicted CorpusIds are compared to the gold mapping after simple string normalization. Per-example accuracy is the fraction of citation masks predicted correctly, and cohort scores are simple averages of these accuracies.

The temporal axis is the citing paper's publication date. For monthly analysis, we recover publication dates from Semantic Scholar metadata. Dates are available for 96.9% of examples and span all 168 months from 2012-01 through 2025-12.

E.9 CLERC

We construct a temporal legal precedent-identification benchmark from CLERC using its CAP-derived corpus (Hou et al., 2025), with decision dates and citation metadata recovered from the Harvard Caselaw Access Project (CAP) (Harvard Law School Library Innovation Lab, 2018). Each example pairs an indirect single-removed (ISR) query with a candidate case. Following CLERC, queries use 150 words on each side of a central citation. The central citation and salient sentence are masked and direct-quotation contexts are excluded. We recover CAP dates and citations by scanning U.S.-jurisdiction volumes and exact-matching CAP IDs to CLERC, recovering 1,838,630 of 1,842,422 documents. We normalize reporter citations using CLERC's `clean_cstr` procedure and keep only aliases that uniquely identify a CLERC document. Citation extraction follows CLERC's eyecite-based code (Cushman et al., 2021).

From 2006–2020, we retain at most one query per cited target in each source case, and exclude targets decided after the source case, yielding 2,691,044 eligible citation events. We drop 2020 as it is incomplete. For each year from 2006–2019, we randomly sample 5,000 eligible citation events. Each event yields one positive example containing the cited case and one hard negative, defined as the highest-ranked BM25-retrieved candidate that is neither the gold nor source case and was decided before the source case. We retrieve over the full corpus using Pyserini 0.24.0/Anserini with BM25 ($k_1 = 0.9, b = 0.4$) (Lin et al., 2021; Yang et al., 2017). The final benchmark contains 140,000 examples, with exactly 5,000 YES and 5,000 NO examples per year.

The model gets the masked passage and candidate text and must return YES/NO. We score accuracy after normalizing predictions to YES or NO, giving invalid outputs a score of zero. The temporal axis is the source case's CAP decision date. Monthly analysis uses the `source_date` field.

F Model Generation Settings

For closed-source models, we evaluate GPT-5.6 Sol and Luna (OpenAI, 2026) under deterministic, tool-free inference. Each benchmark example is supplied with its native prompt and a shared system instruction "You are a helpful assistant." Models receive no web access, retrieval, tools, or information beyond the benchmark. On all datasets, we use temperature 0, a maximum output of 16,384 tokens, no tools, and no response storage. We additionally set reasoning effort to none.

For open-source models, we use 13 instruction-tuned open-weight models spanning four families: Qwen3 (Yang et al., 2025) at 0.6B, 1.7B, 4B, 8B, 14B, and 32B; Gemma-3-IT (Team et al., 2025) at 4B, 12B, and 27B; OLMo-2-1124-Instruct (OLMo et al., 2024) at 7B and 13B; and Llama-3.1-Instruct (Grattafiori et al., 2024) at 8B and 70B. We use the same system instruction as GPT-5.6, with no tools, retrieval, or web access. We decode with temperature 0. For Qwen3, the only family in our panel whose chat template has a thinking toggle, we additionally evaluate each model with thinking on under the recommended sampling settings (temperature 0.6, top-p 0.95, top-k 20). With these 6 thinking variants, we obtain a total of 19 models.

Given the varying sizes and difficulties of our 12 tasks from Appendix E, we set different maximum token settings. In general, we generate up to 32,768 total tokens for non-OLMo models and 4,096 total tokens for OLMo models (their native context length). For some models and tasks, we additionally impose limits on the number of maximum new tokens generated. For thinking variants, we cap generations to 4,096 new tokens on all datasets except BioASQ, StreamingQA, and FinQA, as longer reasoning trajectories are almost always repetitive or unparseable. For non-thinking variants, we set maximum new tokens based on task complexity for larger datasets: 4 for CLERC (a simple yes/no task), 256 for SCIDOCA (citation prediction), and 1,024 for WildTime-arXiv and both NVD tasks (arXiv categories or short cybersecurity answers). These thresholds were determined to be adequate by observing that the fraction of truncated responses remained negligible. The Fakeddit experiments in Appendix K likewise used a maximum of 4 new tokens, due to short yes/no responses.

G WASSERSTEIN MONTHLY LAG ANALYSIS

We quantify the shift in predictive error distributions of 1-lag through 48-lag monthly predictors by plotting the 1-Wasserstein distance between each error distribution and the 1-lag error distribution (Figure 8, right panel). Distance increases steadily with monthly lag, by 0.001 over 48 months, which is 25× larger than the median month-to-month 1-Wasserstein distance. Along with the continuity of the heatmap, this shows that the error distributions of lagged predictors shift smoothly with lag. This smooth distributional shift corresponds to increasing prediction error as evaluations grow stale. We demonstrate this in the left panel of Figure 8, where the 10th through 90th percentiles of absolute error increase nearly linearly with lag (Spearman 0.986–0.997). The upper quantiles show the largest increases, with the 90th percentile prediction error increasing by 0.025. Thus, stale evaluations increase not only average error but also substantially increase tail risk.

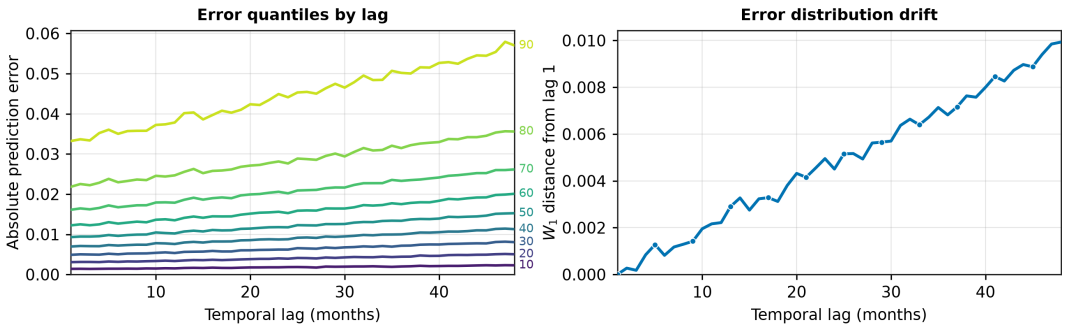

Figure 8: **Prediction error distributions smoothly degrade as evaluations grow stale.** *(Left)* The 10th through 90th percentiles of absolute prediction error increase nearly linearly with monthly lag, with the largest increases occurring in the upper tail. *(Right)* The 1-Wasserstein distance between each lag's error distribution and the 1-lag error distribution increases steadily with monthly lag.

H ANNUAL AND MONTHLY FORECASTING RESULTS

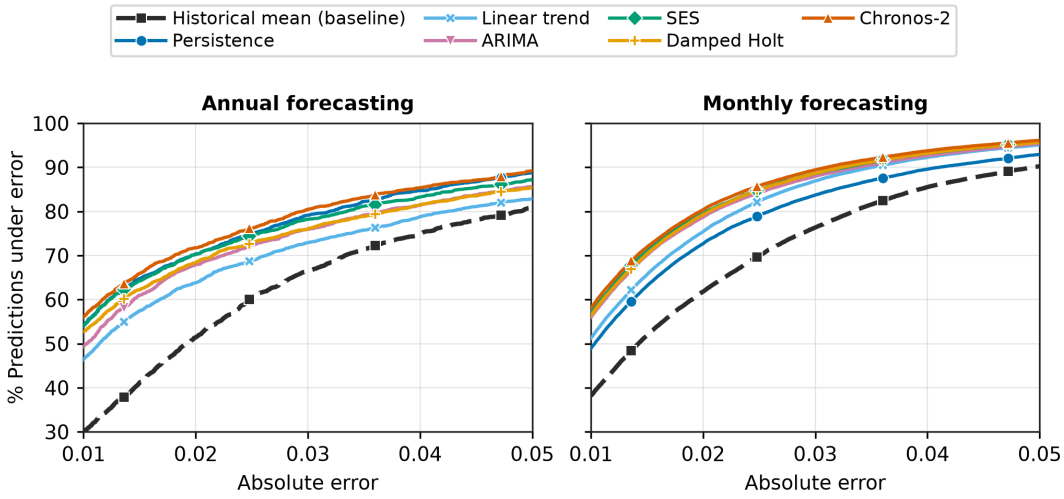

Figure 9: **Time-aware forecasters outperform historical mean pooling, with explicit trajectory modeling providing greater benefits at monthly resolution.** We plot prediction error CDFs for one-step-ahead forecasts at *(left)* annual and *(right)* monthly resolutions. At annual resolution, persistence performs nearly as well as Chronos-2, while at monthly resolution, explicit time-series forecasters like Chronos-2 and SES improve substantially over persistence and perform similarly.

Table 2: **Chronos-2 achieves the lowest MAE at annual and monthly resolutions, although several time-aware forecasters have comparable performance.** Mean absolute error (MAE) is for one-step-ahead forecasts of model performance. Lower is better.

Method	Annual MAE	Monthly MAE
Historical mean (baseline)	0.03083	0.02163
Linear trend	0.02755	0.01527
Damped Holt	0.02360	0.01388
ARIMA	0.02266	0.01425
SES	0.02085	<u>0.01373</u>
Persistence (previous obs.)	<u>0.02001</u>	0.01735
Chronos-2	**0.01938**	**0.01336**

I Model Ranking Inversions

WildTime-arXiv (Yao et al., 2022) provides a concrete example of historical pooling preserving an outdated model ranking (bottom-left panel of Figure 10). Ranking all 19 models by their historical mean selects Llama-3.1-70B-Instruct in all 11 target years, although Qwen3-32B-thinking is the best model in 10 of them. The pooled average remains dominated by evaluations predating the Qwen-Llama crossover, so the historical mean continues to reflect the older ranking. This is the most severe model selection failure due to staleness in our analysis.

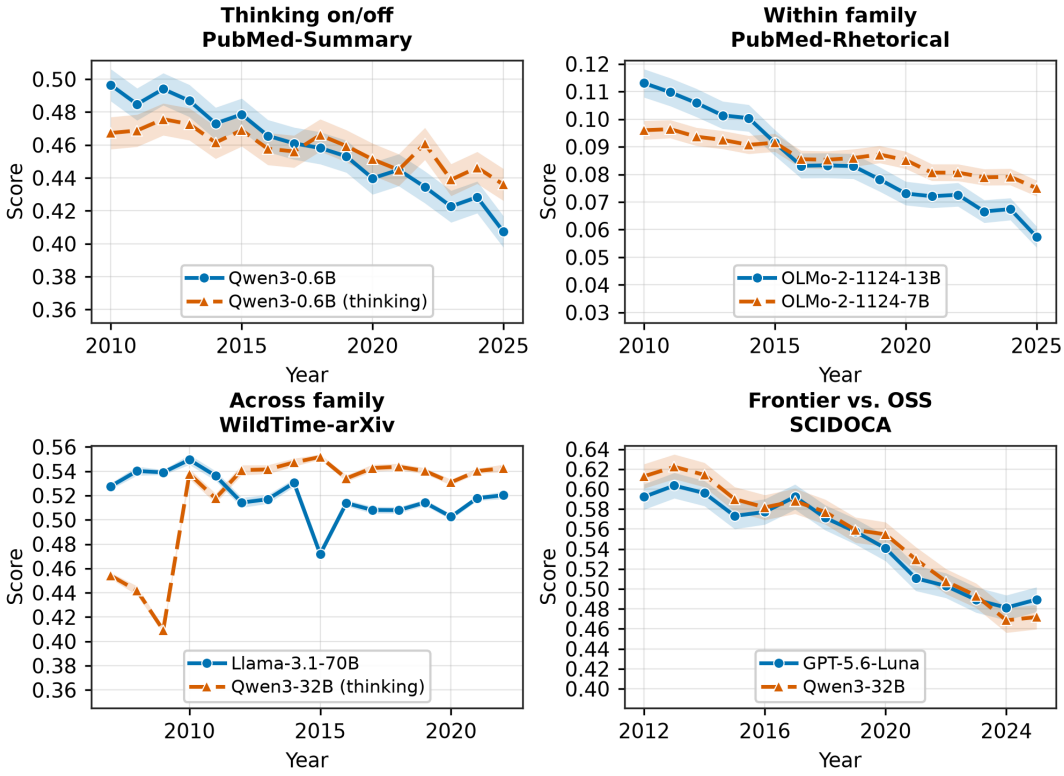

Figure 10: **Model rankings can reverse over time, causing temporally pooled scores to preserve outdated orderings.** Ranking inversions occur across years for thinking versus non-thinking variants, different model sizes within a family, similarly capable models across families, and frontier versus open-weight models. *(Bottom left)* On WildTime-arXiv, the historical mean favors Llama-3.1-70B-Instruct even after Qwen3-32B-thinking becomes the better model in most later years.

J Forecast Error and Model Selection

At annual resolution, persistence and Chronos-2 both improve optimal model selection rates and reduce high-regret choices relative to the historical mean (Figure 11). However, at monthly resolution, persistence is worse than the historical mean on both metrics despite reducing forecast MAE, whereas Chronos-2 improves both metrics relative to the historical mean. We diagnose this discrepancy by decomposing forecast error into a shared component that cannot affect model rankings and a residual component that can affect rankings.

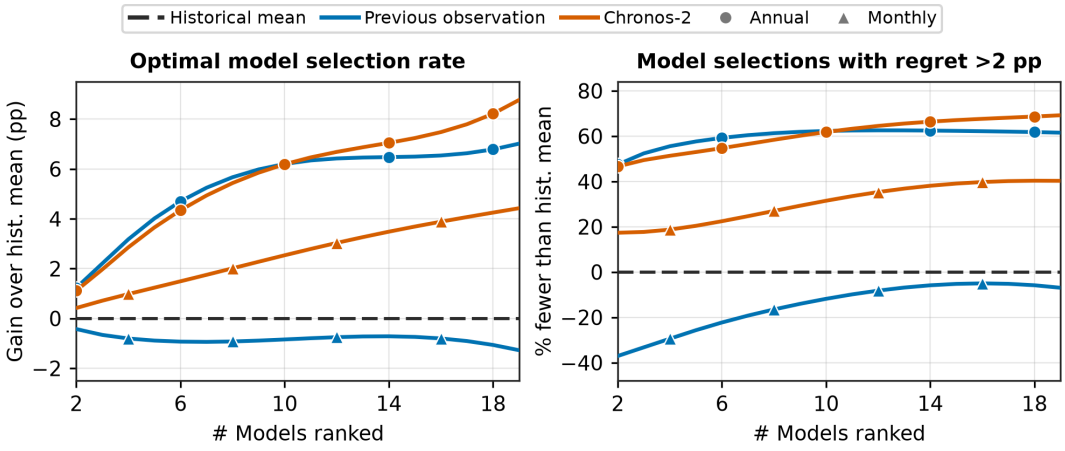

Figure 11: **Chronos-2 improves model selection at both annual and monthly resolutions, while persistence improves only annual selection.** As the number of candidate models increases, we plot *(left)* absolute gains in optimal selection rate over the historical mean and *(right)* reductions in model selections with regret greater than two percentage points relative to the historical mean. Circles are annual results, triangles are monthly results, and the dashed line is the historical mean baseline.

For a dataset b, model m, and target period t, let

$$e_{b,m,t} = \hat{y}_{b,m,t} - y_{b,m,t} \tag{1}$$

denote the forecast error. We decompose this error into the mean error across all $M = 19$ models on the same dataset and target period, and a model-specific residual component:

$$e_{b,m,t} = \bar{e}_{b,t} + \tilde{e}_{b,m,t}, \qquad \bar{e}_{b,t} = \frac{1}{M}\sum_{m=1}^{M} e_{b,m,t}, \qquad \tilde{e}_{b,m,t} = e_{b,m,t} - \bar{e}_{b,t}. \tag{2}$$

Because $\bar{e}_{b,t}$ (the shared error) is identical for every model, removing it cannot change their predicted ordering, including within any shortlist. However, it can affect forecast MAE. This decomposition helps explain the different relationships between forecast error and model selection at annual and monthly resolutions. Annually, persistence's MAE reduction relative to the historical mean falls from 35.1% to 25.5% after removing shared error, while Chronos-2's falls from 37.1% to 23.0%. At monthly resolution, persistence's MAE reduction relative to the historical mean falls from 19.8% to only 3.3%, while Chronos-2 retains a substantial MAE reduction, falling from 38.2% to 23.3%. Thus, much of persistence's monthly reduction in forecast error is shared across models and irrelevant to rankings. This shows why lower forecast MAE does not always translate into a higher optimal selection rate.

K TEMPORAL LOCALITY IN MULTIMODAL EVALUATION

Our experiments thus far are specific to LLMs and text-only benchmarks. We now ask whether the same structure extends to multimodal evaluation, where distribution shift can arise from changes in both visual content and image-text relationships. We evaluate Gemma-3-12B-IT and Gemma-3-27B-IT on Fakeddit (Nakamura et al., 2019), a Reddit task where the goal is to classify pairs of images and post titles as true or fake. We use 10,000 balanced examples per year from 2012–2019.

We again find that recent evaluations are more predictive of future performance than stale ones. At annual resolution, predictive error increases consistently with evaluation age for both models (Figure 12). Because class balance is fixed across time, this pattern reflects changes within the underlying data distribution, such as evolving topics, visual styles, and image-text relationships. Further, all examples predate Gemma 3's August 2024 training cutoff, ruling out post-cutoff evaluation as a cause. Thus, temporal locality extends to multimodal performance, suggesting that it is a general property of evolving evaluation distributions rather than text alone.

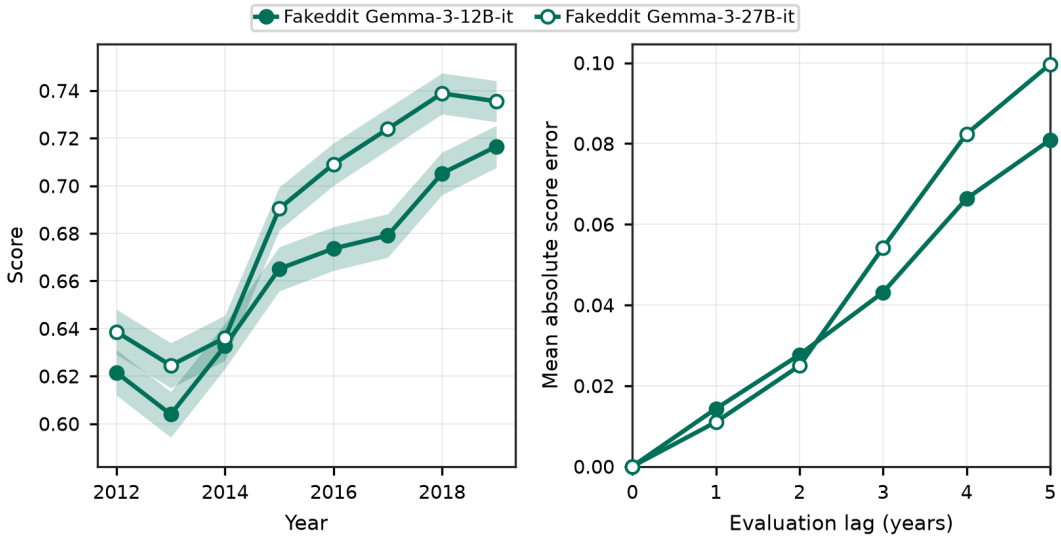

Figure 12: **Recent evaluations better predict future multimodal performance.** We evaluate Gemma-3-12B-IT and Gemma-3-27B-IT on Fakeddit, grouping performance by year. *(Left)* Performance substantially increases over time for both models. *(Right)* Prediction error increases with evaluation age, so recent evaluations are the best predictors of future performance.

L Data Recency in Pretraining

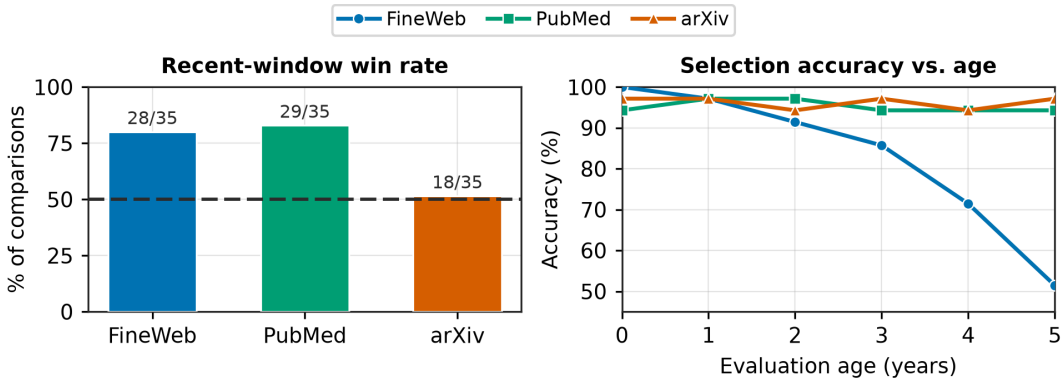

Figure 13: **The benefit of data recency in pretraining is domain-dependent.** *(Left)* Recent-window pretraining often yields lower next-year perplexity than cumulative pretraining on FineWeb and PubMed, while arXiv shows no preference. *(Right)* The most recent yearly target-domain evaluation accurately selects between the two mixtures across all three datasets. Selection accuracy sharply degrades with evaluation age on FineWeb but remains stable on PubMed and arXiv.