What Improves During Pretraining? Tracking Knowledge Acquisition with Point-in-Time Models

Chang Su[α] **Bin Wu**[β] **Emine Yilmaz**[β] **Yao Lu**[β]
[α]University of Pennsylvania
[β]Centre for Artificial Intelligence, University College London
sucha@engineering.upenn.edu bin.wu.23@ucl.ac.uk
emine.yilmaz@ucl.ac.uk yao.lu@cs.ucl.ac.uk

Abstract

Language models pretrained on web corpora collected over multiple years learn both general and domain-specific knowledge. How these different kinds of knowledge are learned as pretraining progresses remains unclear. Recent point-in-time models trained on web snapshots from different years provide a way to probe how knowledge prevalence varies over time. Using these models, we propose a temporal variability score (TVS), defined as the coefficient of variation of a fact's negative log-likelihood across models. A higher score suggests that the knowledge is less consistently represented across years of web text. We evaluate six benchmarks covering diverse topics using six models from multiple open-source model families. We find that low- and high-TVS items are learned similarly early in training, but high-TVS items improve faster later on. At the end of the pretraining run, high-TVS items outperform low-TVS items in 34 of 36 model–benchmark settings, with an average advantage of 7.3 points of chance-normalized accuracy. This pattern is especially pronounced on knowledge-intensive tasks, even when comparing items of similar difficulty. Our findings reveal differences in how knowledge is learned over the course of pretraining, with temporal variability providing a diagnostic of these learning dynamics beyond aggregate benchmark scores.

1 Introduction

Language models improve on a wide range of tasks as pretraining progresses, yet these gains reveal little about the characteristics of the knowledge being learned (Swayamdipta et al., 2020; Chang et al., 2024). We ask how different kinds of knowledge are learned at different stages of pretraining, using their prevalence in training data as a starting point for characterizing these differences.

However, measuring how often a fact or concept appears in pretraining data is difficult because the same knowledge can be expressed in many different ways. Corpus search tools such as infini-gram (Xu et al., 2025) count exact textual matches, but these counts depend on the wording of the query. Passages that express the same knowledge in different words may go uncounted, making exact-match frequency an incomplete measure of prevalence.

The temporal structure of web corpora offers a complementary approach. Pretraining data combine web crawls collected in different years. Our intuition is that knowledge that appears consistently across these years should receive similar likelihoods from models trained on each yearly snapshot, even when its wording differs. Knowledge whose prevalence varies across years may instead receive more variable likelihoods. Point-in-time (PIT) language models, which share an architecture and training recipe but are trained on distinct yearly web snapshots (Drinkall et al., 2024; He et al., 2025; Yan et al., 2026; Kelly et al., 2026), allow us to probe this variation. We propose a temporal variability score (TVS), defined as the coefficient of variation of the negative log-likelihood assigned to an item's question and correct answer across eight yearly PIT-4B models (Kelly et al., 2026). Higher scores suggest that the underlying knowledge appears less consistently across years of web text. TVS measures likelihood variation, not prevalence directly.

TVS enables a fine-grained analysis of knowledge acquisition during pretraining. We use individual items from widely used benchmarks spanning factual knowledge, scientific reasoning, and commonsense reasoning to assess what models learn at different stages. We compute TVS for 16,239 items from six multiple-choice benchmarks, divide each benchmark into length-matched low-, medium-, and high-TVS groups, and track their accuracy across checkpoints of six models at different scales. We examine whether low- and high-TVS items exhibit different learning patterns at different

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 879–891
November 6-10, 2026 ©2026 Association for Computational Linguistics

stages of training.

We find that low- and high-TVS items perform similarly early in training, but high-TVS items improve faster at later stages. At the end of the pretraining runs, high-TVS items outperform low-TVS items in 34 of 36 model–benchmark settings, with an average advantage of 7.3 points of chance-normalized accuracy. The pattern is most robust on knowledge-intensive tasks: on ARC and MMLU, the advantage remains positive in all 18 settings when comparing items of similar difficulty. These findings connect temporal variability to differences in knowledge acquisition that aggregate benchmark scores can mask.

Our contributions are as follows:

- We propose TVS, an item-level diagnostic that uses point-in-time models to characterize sensitivity to training-data composition beyond exact textual matches.
- We document differences in learning across TVS groups: items with higher temporal variability improve more at later training stages, with the most robust evidence on knowledge-intensive tasks.
- We show that, on knowledge-intensive tasks, the component of TVS specific to the correct answer predicts growth in models' preference for that answer during later training stages, among similarly difficult items.

2 Measuring Temporal Variability

Point-in-time models. We use PIT-4B (Kelly et al., 2026), a suite of 4B-parameter base models trained on yearly partitions of FineWeb (Penedo et al., 2024). The eight models for 2017–2024 share architecture, tokenizer, and training setup and differ in the crawl year of their training data. Crawl years index when pages were collected, not when facts originated: each snapshot also contains older, long-lived content. Related PIT and time-stratified models are discussed in §5.

Temporal variability score (TVS). We define the temporal variability score using the coefficient of variation (CV) across training snapshots: for item i, let s_i be the string formed by the question and its gold answer, and let $\mathrm{NLL}_y(s_i)$ be the per-token negative log-likelihood of s_i under the PIT model for year $y \in \{2017, \ldots, 2024\}$. We define

$$\mathrm{TVS}(i) = \frac{\mathrm{std}_y(\mathrm{NLL}_y(s_i))}{\mathrm{mean}_y(\mathrm{NLL}_y(s_i))}. \quad (1)$$

Low TVS means that the yearly models agree on the item relative to its overall likelihood; we call such items *temporally stable*. High TVS means that the item's likelihood depends on the training year; we call such items *temporally variable*. Normalizing by the mean matters. The coefficient of variation of perplexity is strongly entangled with mean perplexity; unnormalized NLL standard deviation mostly selects harder items (Appendix B).

Interpreting TVS. TVS measures relative cross-year variation in likelihood, not prevalence directly or semantic categories such as "commonsense" versus "specialized." Each yearly model scores the same fixed question–answer string: time indexes the training snapshot, not a changing evaluation target. Stable and variable are likelihood-based labels; Table 1 illustrates groups, not a knowledge taxonomy. We also isolate correct-answer variation from the shared question and format (§4.3; Appendix D).

3 Experimental Setup

Datasets. We evaluate on HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), ARC-Easy and ARC-Challenge (Clark et al., 2018), and MMLU (Hendrycks et al., 2021). For each benchmark we sample 3,000 items from the training and validation splits (the test split for MMLU), or use all items when fewer are available, for 16,239 items in total. TVS is correlated with string length, so we stratify within length. We divide each benchmark into ten bins by PIT-4B token length, split each bin into equal low-, medium-, and high-TVS thirds, and pool the bins (Appendix B.2).

Models. We evaluate checkpoints of six models from four open families[1]: Pythia 1.4B and 6.9B (Biderman et al., 2023), TinyLlama 1.1B (Zhang et al., 2024), SmolLM3 3B (Bakouch et al., 2025), and OLMo 1B and 7B (Groeneveld et al., 2024).[2]

Evaluation metric. Each item is scored by ranking its answer options by length-normalized likelihood. We report chance-normalized accuracy:

$$\mathrm{acc}_{\mathrm{norm}} = \frac{\mathrm{acc}_{\mathrm{raw}} - \mathrm{chance}}{1 - \mathrm{chance}}. \quad (2)$$

[1] Checkpoints cover most of each training run, from 0.54B to 11.1T tokens; each trajectory is a single run.

[2] We use the 0724 versions of both OLMo models.

Benchmark	TVS group	Example (question → gold answer)
PIQA	Low	"How do you take soap off of clean dishes?" → "rinse it in water."
PIQA	High	"To tell when baked goods are done baking, you can" → "Wait for them to turn golden brown."
ARC-Easy	Low	"What causes a change in the speed of a moving object?" → "Force."
ARC-Easy	High	"A meteoroid becomes a meteorite when it" → "impacts the surface of a planet."

Table 1: Sampled items from the low- and high-TVS groups.

This maps random guessing to 0 and perfect accuracy to 1 and makes benchmarks with different numbers of options comparable (Yan et al., 2026; Kelly et al., 2026).

4 Results

4.1 Learning Patterns Diverge Across TVS Groups

We examine how learning patterns differ across low- and high-TVS items as training progresses (Figure 1). Our earliest Pythia and OLMo checkpoints occur after 0.54B and 2B training tokens, respectively. At these checkpoints, low-TVS items outperform high-TVS items in 13 of 24 settings, no more often than expected by chance. The mean high-minus-low gap is -0.8 points of chance-normalized accuracy. The groups separate later: over the second half of the checkpoints, accuracy on high-TVS items grows faster by 2.0 points per tenfold increase in training tokens. The difference is positive in 26 of 36 settings and in all six benchmark and model averages, but not significant across four family averages ($p = .13$; Appendix A.1).

At the end of the pretraining runs, high-TVS items outperform low-TVS items in 34 of 36 model–benchmark settings (Table 2). The mean gap is 7.3 points of chance-normalized accuracy (95% CI [5.5, 9.2]). It remains significant when settings are first averaged within each of the four model families (CI [4.8, 10.2], $p = .003$), and it is positive on every benchmark, ranging from 1.7 points on WinoGrande to 16.1 points on PIQA (Appendix A.1). These intervals summarize variation across settings or family averages, not run-to-run uncertainty: each setting contains one training run. The pattern holds across model families trained on different corpora, which suggests that it is not tied to a single training recipe.

4.2 Does TVS Capture More Than Question Difficulty?

To distinguish the relationship between TVS and learning from differences in question difficulty, we repeat the analysis using the variation in TVS not explained by mean NLL. We regress TVS on linear and quadratic mean-NLL terms within each benchmark and length bin, then divide the residuals into thirds. Within each length bin, high-TVS items have slightly lower mean NLL, making this adjustment relevant to the comparison. On ARC and MMLU, the advantage remains positive in all 18 settings, at +6.9 points of chance-normalized accuracy (Table 2). On HellaSwag, PIQA, and WinoGrande, the smaller gap (+0.5 points) indicates a substantial difficulty contribution (Appendix B.1).

	Early	Final		
Benchmarks	L>H	H>L	Gap	Gap | NLL
All	13/24	34/36	+7.3	+3.7
HS/PIQA/WG	8/12	16/18	+8.5	+0.5
ARC/MMLU	5/12	18/18	+6.2	+6.9

Table 2: Low- (L) versus high-TVS (H) items at the earliest and final checkpoints. Early counts cover Pythia and OLMo, whose first checkpoints come early in training; final counts cover all six models. Gaps are high-minus-low chance-normalized accuracy in points. Gap | NLL uses a split conditioned on mean NLL (§4.2).

Conditioning additionally on PIT-4B's gold-versus-distractor margin and accuracy leaves a final gap of 1.4 points, positive in 22 of 36 settings (Table 3). These covariates capture average likelihood and answer discrimination in the diagnostic suite; they need not exhaust difficulty for every evaluated model. The raw 7.3-point gap is not an estimate of an independent TVS effect; the remaining association varies across benchmarks.

We also examine how the results depend on the measure used to group the same items (Table 3). Splitting directly by mean NLL produces the opposite ordering: the higher-NLL group performs worse in all 36 final settings. Replacing TVS with the unnormalized standard deviation of NLL also reverses the effect. This measure increases with mean NLL and therefore selects harder items. These comparisons distinguish the TVS pattern from those obtained by grouping items by mean NLL or unnormalized dispersion.

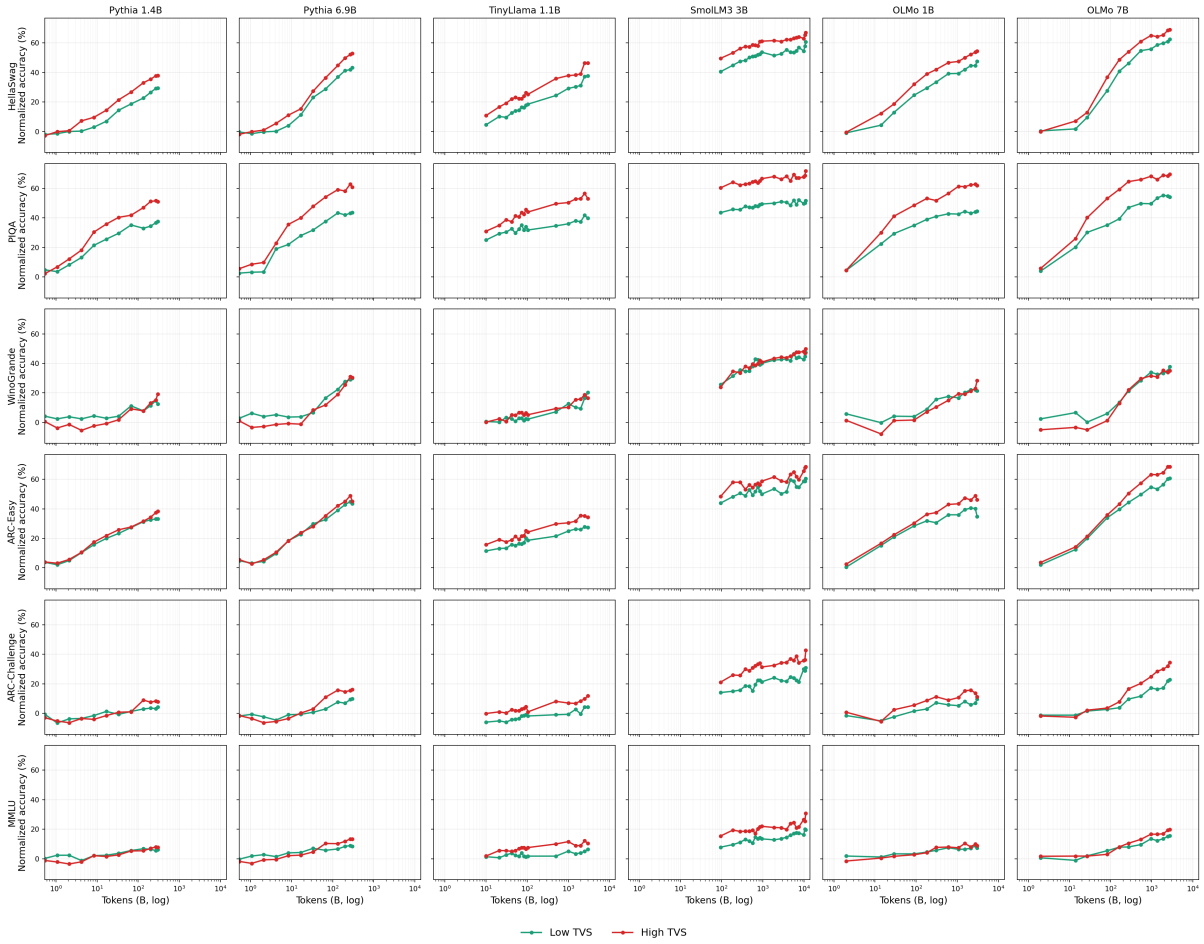


Figure 1: Chance-normalized accuracy over pretraining for temporally variable (high TVS, red) and temporally stable (low TVS, green) items, by model (columns) and benchmark (rows). Groups are length-matched; training tokens are on a log scale. Pythia and OLMo show little early separation; high-TVS items generally pull ahead later.

Split (within length bins)	Final H>L	Gap
TVS (main)	34/36	+7.3
TVS | mean NLL	26/36	+3.7
TVS | mean NLL, PIT margin, PIT acc.	22/36	+1.4
Gold-vs.-wrong TVS margin	36/36	+12.3
Question-only TVS	26/36	+2.1
NLL standard deviation	8/36	−2.6
Mean NLL (high = harder)	0/36	−31.1

Table 3: Final-checkpoint comparison of high- versus low-group items under alternative stratifications (chance-normalized points; 36 model–benchmark settings). Conditioned splits use the residual of TVS after regressing out the listed covariates within each benchmark and length bin. Full results are in Table 5.

4.3 Answer-Specific TVS Predicts Learning in Later Training Stages

Answer options share a question and format, so we isolate correct-answer variation as its TVS minus the mean TVS of incorrect answers. We test whether it predicts growth in the correct-answer-versus-best-distractor margin, measured as its slope against log training tokens over the upper half of the log-token range. Margin growth need not change the top-ranked option, so we analyze it separately from final accuracy. Regressions include shared and answer-specific components, controlling for PIT-4B mean NLL, margin, and accuracy. The answer-specific signal predicts greater late margin growth on ARC-Challenge, ARC-Easy, and MMLU ($p < 10^{-6}$ each; Table 4), with a weaker effect on PIQA. On HellaSwag and WinoGrande, it predicts a small final-accuracy difference but no robust late-margin growth (Appendix D).

Group comparisons agree: the correct-answer-versus-strongest-distractor TVS margin yields the largest late-stage gap, with the high group ahead in all 36 settings by 12.3 points on average and the best final group in 35. In contrast, question-only TVS yields an average gap of just 2.1 points (Appendices C.2 and C.1). Thus the relationship

Benchmark	Final correct	Late margin slope
ARC-C	+0.024 (.04)	+0.070 (2×10^{-9})
ARC-E	+0.008 (.2)	+0.059 (2×10^{-8})
MMLU	+0.001 (.8)	+0.043 (2×10^{-7})
PIQA	+0.001 (.8)	+0.011 (2×10^{-4})
HellaSwag	+0.019 (10^{-5})	0.000 (.8)
WinoGrande	+0.012 (.01)	+0.002 (.08)

Table 4: Difficulty-controlled effect of gold-specific TVS (per standard deviation) on final correctness and late gold-versus-distractor margin growth (p-values in parentheses; Appendix D).

depends substantially on the answer itself, rather than only on question wording.

4.4 Assessing TVS Robustness to Benchmark Contamination

We test for benchmark exposure in PIT-4B using release dates and textual overlap (Appendix F). Benchmark copies should raise gold-answer likelihood after release. Yet staggered difference-in-differences with item and year fixed effects finds no decrease in gold-answer NLL ($+0.016$) or detectable change in the gold-versus-distractor margin ($p = .34$). High-TVS accuracy falls slightly (-0.011, $p = .004$), opposite the expected leakage effect. ARC, PIQA, and MMLU share similar year profiles despite different release dates, with no release-specific gains.

We also scan 55B FineWeb tokens, using ten files per crawl year, for 13-gram overlap with each item (Brown et al., 2020). Overlap is rare (3.1% of high-TVS versus 2.0% of low-TVS items) and mostly comes from source pages such as WikiHow rather than answer keys. Years with more overlap do not have lower PIT-4B NLL for the matched items. Adding overlap indicators to the item-level regressions leaves the TVS coefficient essentially unchanged. These checks provide evidence against benchmark leakage as the dominant explanation of the TVS pattern in PIT-4B, but sparse yearly sampling cannot establish its absence or exclude contamination of the evaluated models' corpora.

5 Related Work

Benchmark Analysis. Dataset cartography characterizes items through a fine-tuned model's training dynamics (Swayamdipta et al., 2020), while item response theory estimates their difficulty and discrimination (Lalor et al., 2016; Martínez-Plumed et al., 2019). Other approaches group benchmark items by difficulty or composition (Ding et al., 2024; Kovatchev and Lease, 2024). These measures describe items through the behavior of the evaluated models and can therefore vary with the model being diagnosed. TVS instead uses an external suite that shares architecture and training setup but differs in its temporal data slice. It therefore characterizes items independently of the evaluated model's own training trajectory and supports comparisons across model families.

Attributing Learning to Pretraining Data. Models learn facts through repeated exposure (Chang et al., 2024), memorize more over training (Tirumala et al., 2022), and struggle with rare knowledge (Kandpal et al., 2023). Corpus search counts exact matches (Elazar et al., 2024; Xu et al., 2025), making exposure estimates wording-dependent. Controlled ablations link capabilities to corpus components: removing parallel bilingual documents impairs translation but largely preserves cross-lingual question answering and reasoning (Shao et al., 2026). TVS complements these approaches by measuring sensitivity beyond exact overlap without requiring corpus interventions.

Point-in-Time Language Models. Time-stratified models probe temporal generalization and time-sensitive facts (Lazaridou et al., 2021; Dhingra et al., 2022); PIT suites also address lookahead bias and historically bounded knowledge (Drinkall et al., 2024; He et al., 2025; Yan et al., 2026; Kelly et al., 2026). We instead characterize fixed benchmark items through yearly likelihood variation, defining groups independently of evaluated checkpoints and comparing trajectories without their training corpora. Time indexes the corpus, not an intervention or changing targets.

6 Conclusion

We introduced TVS to track item-level learning using point-in-time models. Across six benchmarks and six models, high-TVS items generally improve more later; the earliest Pythia and OLMo checkpoints show little group separation. Effects are most robust on ARC and MMLU, where difficulty-controlled, answer-specific TVS predicts late margin growth. TVS reveals dynamics beyond aggregate scores without requiring training-corpus access or assigning semantic categories to items.

Limitations

TVS is a model-based likelihood statistic, and its effectiveness therefore depends on the quality of the underlying model(s). The PIT-4B models cover crawl years 2017–2024 at a single model size. Our contamination checks cover the PIT-4B training data rather than the full pretraining corpora of the evaluated models (Appendix F). Finally, we study only likelihood-ranked multiple-choice benchmarks as they are most suitable when monitoring models' early checkpoints.

Acknowledgments

This research is supported by the Engineering and Physical Sciences Research Council [grant number UKRI5028] and a 2025 Amazon Research Award. We also gratefully acknowledge NVIDIA Corporation for their support through the NVIDIA Academic Grant Program, which provided GPU computing resources used in this research. Part of this research was supported by the Bloomberg Data Science PhD Fellowship programme.

References

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, and 4 others. 2025. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. How do large language models acquire factual knowledge during pretraining? In *Advances in Neural Information Processing Systems*, volume 37, pages 60626–60668. Curran Associates, Inc.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Anima Anandkumar, and Furong Huang. 2024. Easy2Hard-Bench: Standardized difficulty labels for profiling LLM performance and generalization. In *Advances in Neural Information Processing Systems*, volume 37, pages 44323–44365. Curran Associates, Inc.

Felix Drinkall, Eghbal Rahimikia, Janet Pierrehumbert, and Stefan Zohren. 2024. Time machine GPT. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3281–3292, Mexico City, Mexico. Association for Computational Linguistics.

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's in my big data? In *The Twelfth International Conference on Learning Representations*.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Bryan T. Kelly, Semyon Malamud, Johannes Schwab, and Teng Andrea Xu. 2026. Scaling point-in-time language models. Research Paper 26-37, Swiss Finance Institute.

Venelin Kovatchev and Matthew Lease. 2024. Benchmark transparency: Measuring the impact of data on evaluation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1536–1551, Mexico City, Mexico. Association for Computational Linguistics.

John P. Lalor, Hao Wu, and Hong Yu. 2016. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 648–657, Austin, Texas. Association for Computational Linguistics.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2021. Mind the gap: Assessing temporal generalization in neural language models. In *Advances in Neural Information Processing Systems*, volume 34, pages 29348–29363. Curran Associates, Inc.

Fernando Martínez-Plumed, Ricardo B. C. Prudêncio, Adolfo Martínez-Usó, and José Hernández-Orallo. 2019. Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artificial Intelligence*, 271:18–42.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pages 30811–30849. Curran Associates, Inc.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WinoGrande: An adversarial Winograd schema challenge at scale. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Jiandong Shao, Raphael Tang, Crystina Zhang, Karin Sevegnani, Pontus Stenetorp, Jianfei Yang, and Yao Lu. 2026. The role of mixed-language documents for multilingual large language model pretraining. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36807–36818, San Diego, California, United States. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 38274–38290. Curran Associates, Inc.

Hao Xu, Jiacheng Liu, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. Infini-gram mini: Exact n-gram search at the Internet scale with FM-index. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24944–24969, Suzhou, China. Association for Computational Linguistics.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. TinyLlama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

A Additional Results

Table 5 compares the main split with alternative stratifications. For each model–benchmark setting we compare the earliest and final checkpoints. "Early L>H" counts settings in which the low group is ahead at the earliest checkpoint; unlike Table 2, it includes all six models. "Final H>L" counts the reverse at the final checkpoint. "Best H" counts settings in which the high group has the highest final accuracy of all groups. No setting had exactly tied accuracies. Figure 2 shows the main result with five TVS quantiles instead of three.

A.1 Effect Sizes and Significance

All values are in chance-normalized accuracy, and each model–benchmark setting has one training run. The mean final high-minus-low gap across the 36 settings is +0.073 (95% t-CI [0.055, 0.092], $p = 1.9\times10^{-9}$). The conclusion does not depend on the unit of analysis. Averaging first within benchmarks gives [0.023, 0.124] ($p = .014$); within models gives [0.059, 0.088] ($p = 4.9 \times 10^{-5}$); and within the four families (Pythia, TinyLlama, SmolLM3, OLMo) gives +0.075, [0.048, 0.102] ($p = .003$). Averaged over models, the final gap is 0.070 on ARC-C, 0.068 on ARC-E, 0.078 on HellaSwag, 0.046 on MMLU, 0.161 on PIQA, and 0.017 on WinoGrande. Averaging the last two or three checkpoints instead of the final one gives the same result.

At the earliest checkpoints of Pythia and OLMo, the mean gap is -0.008 (95% CI [-0.018, 0.003]); the low group is ahead in 13 of 24 settings (binomial $p = .84$). To measure how the groups diverge, we fit, for each setting, an ordinary least-squares slope of accuracy on $\log_{10}$ training tokens over the second half of the checkpoints, ordered by training step. The high-group slope exceeds the low-group slope by 0.020 per tenfold increase in tokens (positive in 26/36 settings; 95% CI [0.011, 0.029]). The difference is positive for all six benchmark means (CI [0.010, 0.030]) and all six model means (CI [0.001, 0.039]), but not significant across the four family means (+0.016, $p = .13$).

B Controls

B.1 Mean Difficulty

Within length bins, high-TVS items have lower mean NLL in every benchmark. The within-bin Spearman correlations between TVS and mean NLL are -0.08 (ARC-C), -0.05 (ARC-E), -0.24 (HellaSwag), -0.10 (MMLU), -0.24 (PIQA), and -0.10 (WinoGrande). The corresponding high-minus-low differences in mean NLL are -0.03, -0.02, -0.19, -0.11, -0.32, and -0.13 nats (Table 6).

Because mean NLL strongly predicts accuracy, we also split on the residual of TVS after regressing it on mean NLL within each benchmark and length bin. This difficulty-conditioned split roughly halves the final gap, from +0.073 (34/36) to +0.037 (26/36). What remains is concentrated in ARC-C (0.075), ARC-E (0.076), MMLU (0.056), and PIQA (0.050), with none on HellaSwag (-0.014) or WinoGrande (-0.023). Additionally conditioning on the PIT-4B gold-versus-distractor margin and PIT-4B accuracy leaves +0.014 (22/36).

The choice of dispersion statistic also matters. The unnormalized standard deviation of NLL reverses the pattern (-0.026, 8/36): it increases with mean NLL (within-bin Spearman $\rho = 0.26$) and therefore selects harder items. A rank-based measure of disagreement, which is nearly uncorrelated with mean NLL, gives a weak positive gap (+0.011, 22/36).

B.2 Length

Raw TVS decreases with string length, so we form all groups within length bins. For each benchmark, we sort items by the PIT-4B token length of the gold string, divide them into ten bins, rank items by TVS within each bin, and assign low, medium, and high groups within the bin before pooling. After this stratification, the correlation between TVS and length is essentially zero (Pearson -0.016, Spearman 0.006; Table 6), so the high-TVS advantage cannot be attributed to high-TVS items being shorter or longer.

B.3 Perplexity-Based CV

An alternative is the coefficient of variation of perplexity rather than that of NLL. Because perplexity exponentiates NLL, its dispersion grows with its level: $\mathrm{CV}_{\mathrm{PPL}}$ correlates at about 0.43–0.49 with mean perplexity and mean NLL (Table 6), which makes it hard to separate temporal variation from difficulty. We therefore use TVS throughout.

C Alternative Target Strings

We compute TVS on the question together with its gold answer, because the question alone often underspecifies the relevant fact or procedure. Two alternative targets test this choice.

Stratification	Early L>H	Final H>L	Best H	Mean final H−L
Gold TVS tertiles (main)	15/36	34/36	28/36	+0.073
Gold TVS quintiles	15/36	32/36	25/36	+0.081
Mean gold NLL quintiles	35/36	0/36	0/36	−0.311
Question-only TVS tertiles	13/36	26/36	20/36	+0.021
Gold-vs.-wrong TVS margin tertiles	20/36	36/36	35/36	+0.123
Gold NLL-SD tertiles	30/36	8/36	7/36	−0.026
Rank-disagreement tertiles	15/36	22/36	16/36	+0.011
TVS | mean NLL tertiles	18/36	26/36	22/36	+0.037
TVS | mean NLL, PIT margin, PIT acc. tertiles	21/36	22/36	18/36	+0.014

Table 5: Alternative stratifications across 36 model–benchmark settings. Mean final H−L is the final-checkpoint high-minus-low gap in chance-normalized accuracy. For mean gold NLL, "high" is the higher-NLL group; for the gold-vs.-wrong margin, "high" has the larger $\mathrm{TVS}(q, a^{\mathrm{gold}}) - \mathrm{TVS}(q, a^{\mathrm{wrong}})$. Rows below the rule use the same ten length bins. NLL-SD is the unnormalized standard deviation of gold NLL across years. Rank disagreement is the standard deviation across years of an item's NLL percentile within its benchmark and length bin. The conditioned rows split on the residual of TVS after regressing it (with quadratic terms) on the listed covariates within each benchmark and length bin (Appendix B.1).

Signal	Control	Pearson	Spearman
$\mathrm{CV}_{\mathrm{PPL}}$	log mean PPL	0.486	0.426
$\mathrm{CV}_{\mathrm{PPL}}$	mean NLL	0.443	0.423
$\mathrm{CV}_{\mathrm{PPL}}$	length	−0.169	−0.331
$\mathrm{CV}_{\mathrm{PPL}}$	log length	−0.223	−0.406
TVS	log mean PPL	0.166	0.011
TVS	mean NLL	0.134	0.007
TVS	length	−0.119	−0.159
TVS	log length	−0.142	−0.205
Length-controlled TVS	length	−0.016	0.006
Length-controlled TVS	log length	−0.011	−0.004
Length-controlled TVS	mean PPL	0.011	0.014
Length-controlled TVS	mean NLL	−0.079	−0.091

Table 6: Correlations (with benchmark fixed effects) between the two dispersion measures (TVS and perplexity-based CV) and simple item properties over all 16,239 items. Length is the PIT-4B token length of the gold string. $\mathrm{CV}_{\mathrm{PPL}}$ is strongly entangled with mean difficulty and length. TVS is much less related to mean difficulty but still correlates with length, which motivates length-stratified grouping; after length control, correlations with length and mean PPL are near zero.

C.1 Question-Only TVS

Computing TVS on the question alone yields a much weaker effect: high-TVS items outperform low-TVS items in 26 of 36 final settings, by 0.021 on average, compared with 0.073 for the question and gold answer together (Figure 4). The signal therefore depends substantially on the correct answer and not only on the wording of the question.

C.2 Gold Answer versus Best Wrong Answer

To isolate variability specific to the correct answer, we subtract the TVS of the best wrong answer from that of the gold answer:

$$\Delta\mathrm{TVS}(i) = \mathrm{TVS}(q_i, a_i^{\mathrm{gold}}) - \mathrm{TVS}(q_i, a_i^{\mathrm{wrong}}). \quad (3)$$

Splitting on this margin gives the strongest late-stage separation of any stratification. The high-margin group is ahead in all 36 final settings and is the best group in 35 of them, with a mean gap of 0.123 (Table 5, Figure 5).

D Item-Level Analysis by Benchmark

We decompose TVS into a component shared with the distractors (the mean TVS over wrong-answer strings) and a gold-specific component (gold minus distractor TVS). For each benchmark, we regress two item-level outcomes on both components. The first is correctness at the final checkpoint. The second is late-training margin growth: the least-squares slope of the evaluated model's gold-versus-best-distractor margin (in mean NLL) on $\log_{10}$ tokens, over checkpoints in the upper half of the log-token range. Each regression controls for PIT-4B mean gold NLL, the PIT-4B gold-versus-best-distractor margin, PIT-4B accuracy, and fixed

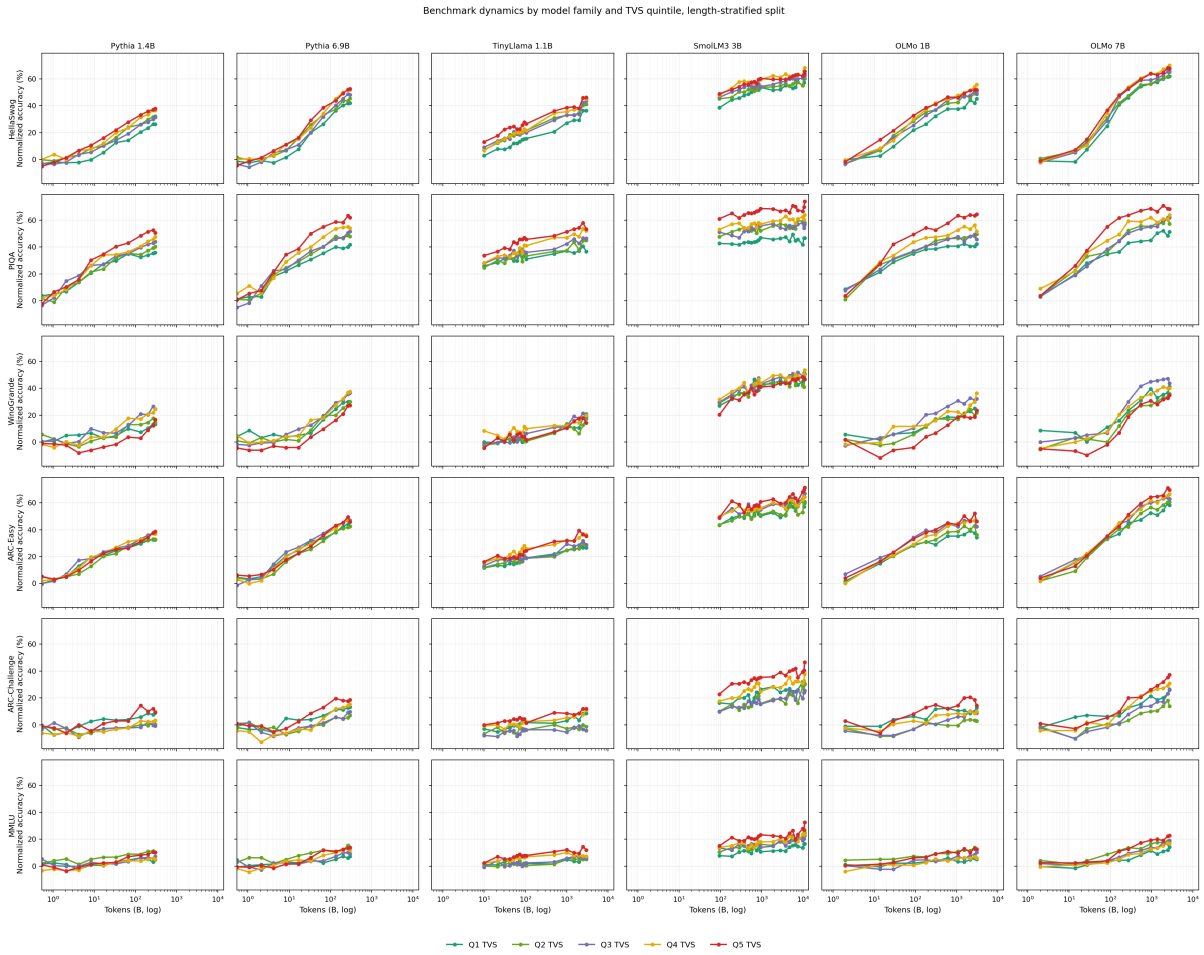


Figure 2: Main result with five TVS quantiles. The advantage of the most variable items at late checkpoints persists under this finer split.

effects for length bin and model, with standard errors clustered by item. Table 4 reports effects per standard deviation of the gold-specific component.

The gold-specific component strongly predicts late margin growth on ARC-C, ARC-E, and MMLU, weakly on PIQA, and not at all on HellaSwag or WinoGrande, where it instead predicts a small difference in final correctness. The benchmarks also differ in how TVS is composed. On the three benchmarks posed as "Question: … Answer:" with short answers (ARC-C, ARC-E, MMLU), 90–92% of the variance in gold TVS is shared with the distractors (correlation 0.93–0.96), compared with 53–59% on HellaSwag and WinoGrande. The difference between benchmarks may therefore reflect answer format as well as knowledge type.

E Additional Experiments on What Does TVS Measure

Synthetic intervention. For each of five seeds, we trained eight small causal language models, one per synthetic "era." Token, entity, and answer counts and total support were held fixed across eras; only the allocation of each answer's conditional support over eras differed. Making support era-dependent raised TVS from 0.229 to 0.351 (5/5 seeds; paired $t(4) = 11.2$, $p = 3.6 \times 10^{-4}$), whereas a stable 50/50 mixture of conflicting evidence did not raise it. The training length (6 epochs) was chosen after a single-seed 12-epoch pilot in which the main contrast was weaker (+0.156, 9/16 concepts, $p = .09$). A contrast varying only surface form was inconclusive (+0.037, $p = .42$; -0.48 in the pilot). The experiment shows that TVS *can* reflect era-varying conditional support, not that high-TVS benchmark items arise this way.

Training-data frequency. Using 25.7B tokens sampled from the training mixture of OLMo-7B-0724, we counted 2- and 3-grams ending in each gold answer and the corresponding gold-versus-distractor frequency margins. We then regressed the change in OLMo-7B-0724's gold margin be-

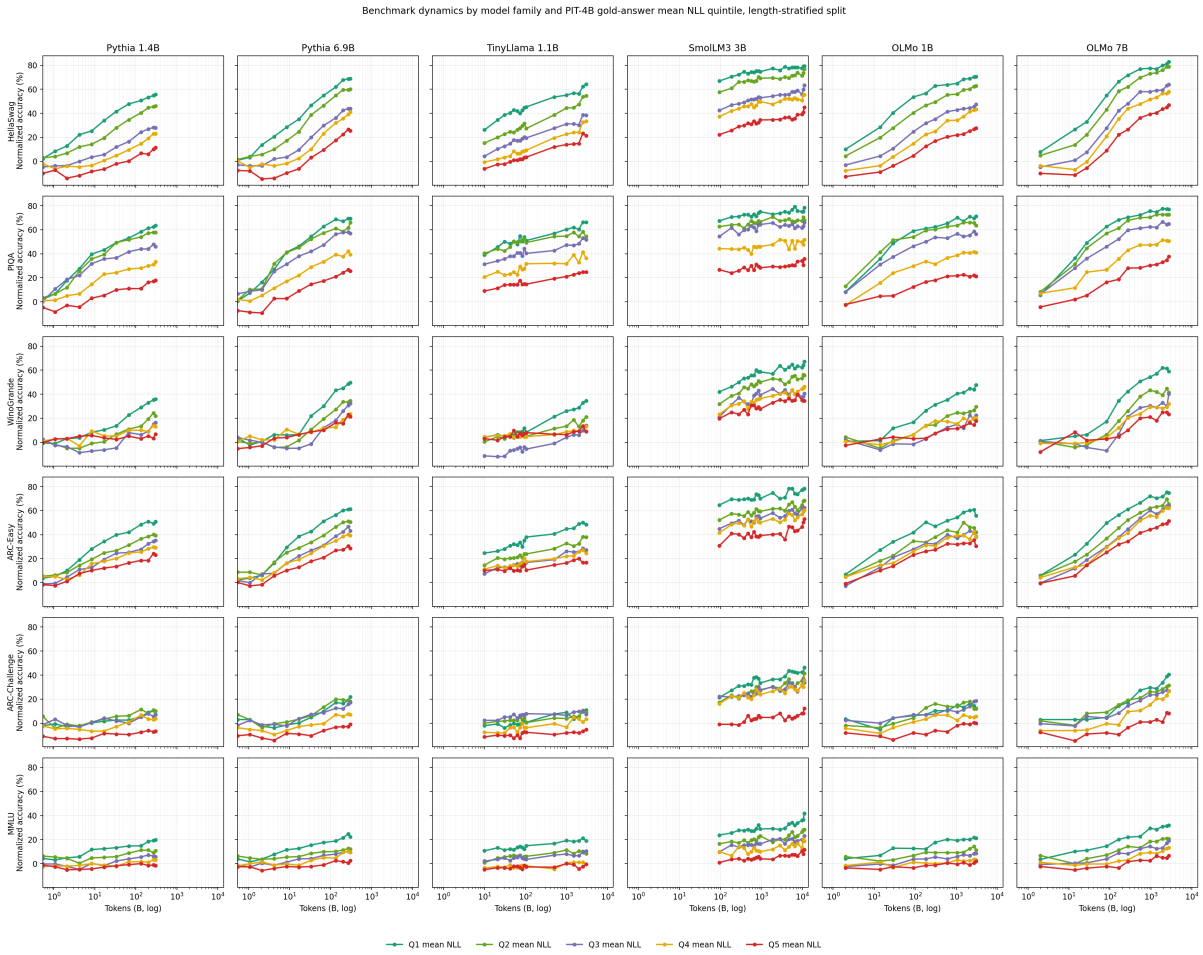


Figure 3: Split by mean gold NLL. Higher mean NLL is associated with lower accuracy at all checkpoints. Difficulty-conditioned TVS results are reported separately in Appendix B.1.

tween 83B and 2457B training tokens on these counts and TVS. Low answer frequency is the dominant predictor (partial $R^2 \approx 0.021$). TVS retains a small independent association ($\beta = 0.031$ per SD, 95% CI [0.017, 0.045], HC3 standard errors; $\Delta R^2 \approx 0.001$) that is driven by ARC-C, ARC-E, and MMLU and varies across benchmarks ($I^2 = 74\%$). N-gram counts capture surface wording rather than exposure to the underlying fact.

F Contamination Checks

Release-date test. If copies of benchmark items entered later crawls, PIT-4B models trained on crawls from after a benchmark's release should assign its gold answers lower NLL than the year trend shared by all benchmarks would predict. We take the release years to be 2018 for ARC, 2019 for HellaSwag and WinoGrande, 2020 for PIQA, and 2021 for MMLU. We estimate a staggered difference-in-differences model with item and year fixed effects. Post-release gold NLL is slightly *higher* (+0.016), with no detectable gold-versus-distractor margin change ($p = .34$), and PIT-4B accuracy on high-TVS items falls (-0.011, $p = .004$). ARC, PIQA, and MMLU also share the same year profile despite their different release years. This test finds no release-specific improvement predicted by benchmark leakage into PIT-4B's training snapshots.

Verbatim overlap. Following Brown et al. (2020), we scanned 55.4B tokens of FineWeb, ten files per crawl year from 2017 to 2024, for 13-gram overlaps with each item. We searched separately for the question alone, for 13-grams spanning the question and the gold answer, and for the gold answer alone, using the same spans for wrong answers as controls. Overlap is rare and mostly comes from source text, such as WikiHow pages, rather than from pages that pair questions with their answers. Any overlap occurs for 3.1% of high-TVS and 2.0% of low-TVS items (length-stratified OR $= 1.56$, $p = 3 \times 10^{-4}$). For gold-answer text the figures are 2.1% versus 1.0%

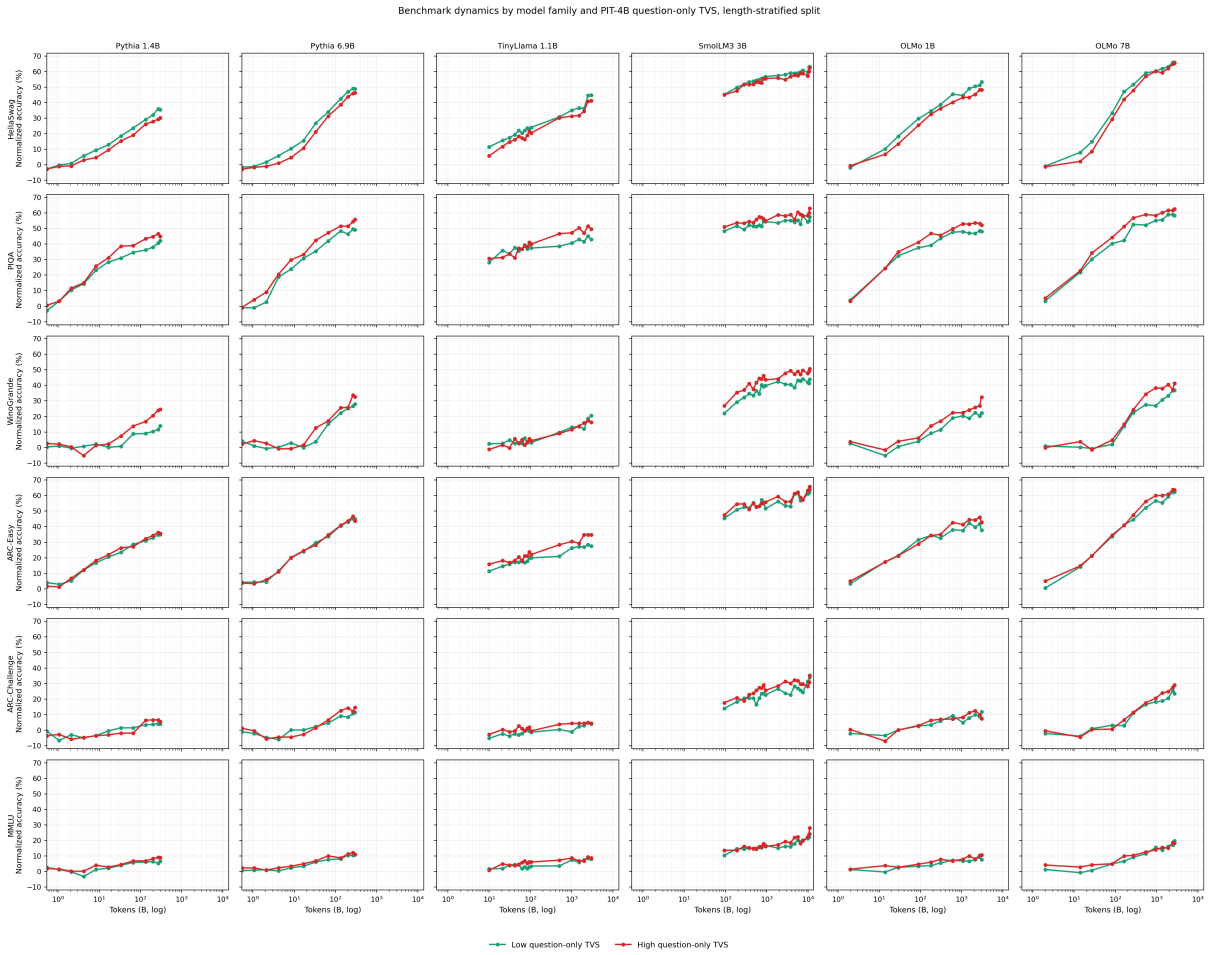


Figure 4: Split by question-only TVS. The effect is much weaker than for the question and gold answer together, so the signal is not reducible to the question stem.

(OR $= 1.90$), and for distractor text 0.6% versus 0.3% (OR $= 1.56$). Within items, years with more overlap do not have lower PIT-4B NLL (question: $p = .63$; question–gold span: $p = .60$; gold answer: $p = .08$). Adding overlap indicators to the item-level regressions leaves the TVS coefficient essentially unchanged (0.0267 versus 0.0268 for final correctness). These checks provide evidence against verbatim overlap as the dominant explanation, but sparse yearly sampling cannot establish the absence of contamination.

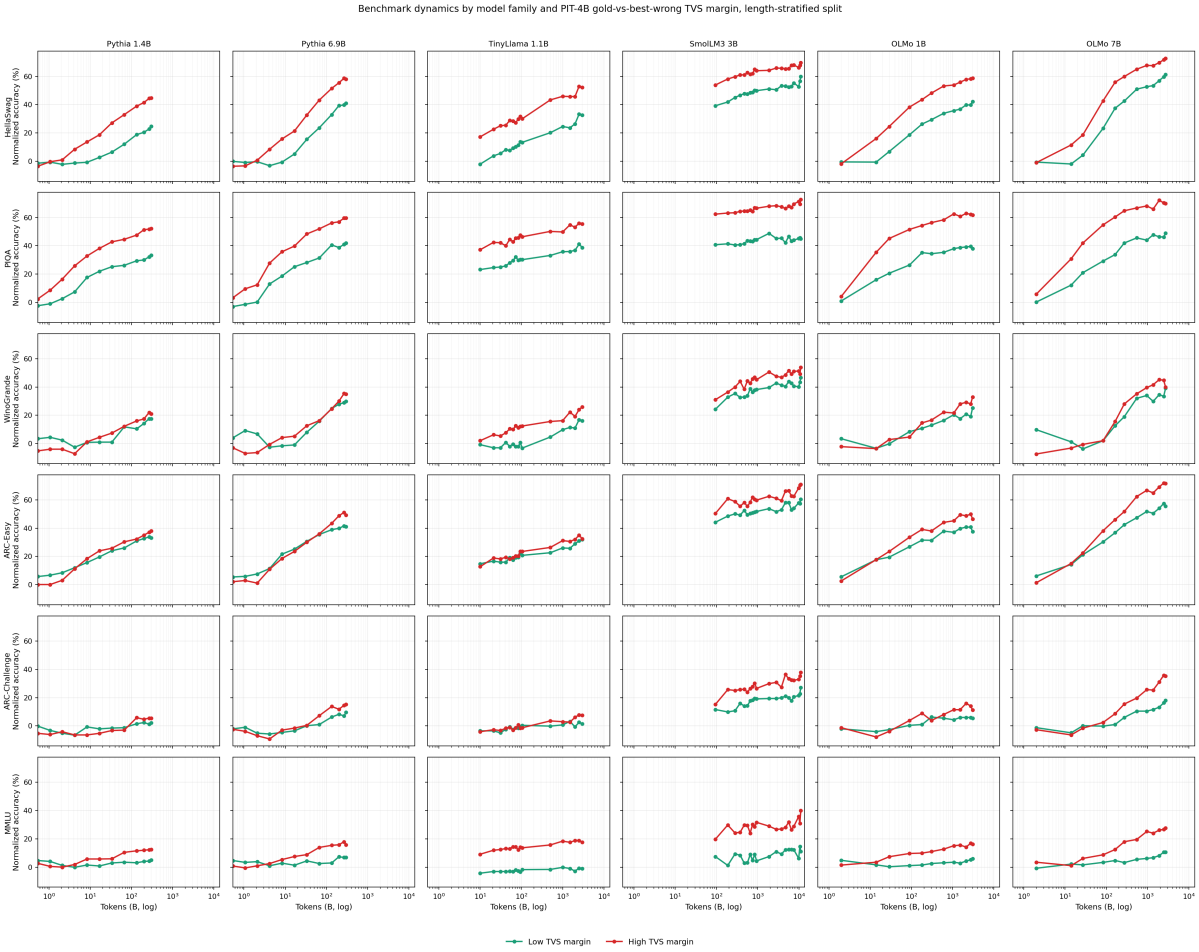


Figure 5: Split by the gold-versus-wrong TVS margin. Items whose gold answer is more temporally variable than the best wrong answer show the strongest late-stage advantage.