Memorization Dynamics of Fill-in-the-Middle Pretraining

Tobias von Arx* [1] **Tanguy Dieudonné*** [1]

Abstract

Fill-in-the-middle (FIM) is a pretraining objective widely used to equip causal language models with infilling ability, yet its effect on verbatim memorization remains underexplored. We study the memorization dynamics of FIM in a controlled setting by pretraining matched Llama 3.2 models with FIM and standard left-to-right (LTR) objectives on a FineWeb-Gutenberg corpus containing repeated Gutenberg excerpts. With prefix-based probes, FIM more often recovers short or partially matching spans, while LTR more often assigns high confidence to long exact continuations. We observe that verbatim extraction under FIM-training grows approximately linearly with repetitions over the tested range. Evaluating native FIM-format probes reveals that suffix context is not sufficient: verbatim recall under FIM-training remains strongly anchored in prefix context. Our results also show that evaluating only one span length or probing format can miss important nuances in memorization behavior.

1. Introduction

Large language models can reproduce training data, including rare strings, private information, code, and book passages (Carlini et al., 2019; 2021; Nasr et al., 2025; Cooper et al., 2026). Early work measured unintended memorization with synthetic canaries and exposure scores (Carlini et al., 2019); later attacks extracted real training examples (Carlini et al., 2021; Nasr et al., 2025). Recent work studies leakage beyond greedy decoding, including probabilistic extraction (Hayes et al., 2025), book-level extraction (Cooper et al., 2026), and membership-style tests (Mattern et al., 2023; Shi et al., 2024).

*Equal contribution [1]Department of Computer Science, ETH Zurich, Zurich, Switzerland. Correspondence to: Tobias von Arx <tvonarx@ethz.ch>, Tanguy Dieudonné <tdieudonne@ethz.ch>.

Published at ICML 2026 Workshop on the Impact of Memorization on Trustworthy Foundation Models, Seoul, South Korea.

Repetition is one of the clearest predictors of memorization. Deduplication reduces verbatim generations (Lee et al., 2022); duplicate count predicts regeneration (Kandpal et al., 2022); and controlled injections are recovered more often as exposure increases (Huang et al., 2024). Attribution remains difficult because prior predictability, near duplicates, tokenization, prompt position, and available context can all affect recovery (Kharitonov et al., 2021; Zhang et al., 2023; Shilov et al., 2026; Liu et al., 2024; Xu et al., 2026).

We study fill-in-the-middle (FIM), a common pretraining objective for causal language models (Bavarian et al., 2022). Standard left-to-right (LTR) training predicts each token from its prefix. FIM training moves a target middle span after prefix and suffix, separated by sentinel tokens, such that during training, the target is exposed to right context as well as left context. Infilling is used in systems such as DeepSeek-v3, InCoder, StarCoder, and Code Llama (DeepSeek-AI et al., 2025; Fried et al., 2023; Li et al., 2023; Rozière et al., 2023). Prior work has mainly emphasized infilling utility; here we ask how the objective impacts verbatim extraction.

We conduct a controlled study comparing standard LTR and FIM pretraining under matched architecture and data source, asking three related questions:

(i) How does FIM impact verbatim memorization across target span lengths, extraction thresholds, and repetition?

(ii) Under native FIM prompting, how do prefix context, suffix context, and sentinel tokens contribute to verbatim memorization?

(iii) Are the observed effects specific to extraction geometry, or explained by broad model-quality differences?

2. Study Design

We compare paired LTR and FIM models trained on the same data, architecture and parameters. Our controlled conditions let us attribute differences in memorization to the pretraining format.

We release our code at `https://github.com/tobiasvonarx/memorization-study-fim`.

2.1. Matched Training with Controlled Repetition

The bulk corpus is FineWeb 100B, while our controlled memorization corpus consists of Project Gutenberg books (Penedo et al., 2024; Project Gutenberg, n.d.). We score 4096-token windows of Gutenberg books with a Llama 3.2 model (Llama Team, 2024) trained only on FineWeb, in order to filter out pre-memorized, outlier, and duplicate windows. The resulting cleaned set of excerpts is split into 12 repetition buckets of 2,810 excerpts with exposures from 1 to 128. We balance bucket assignment by prior perplexity.

We build two corpora from the same data sources. The LTR corpus keeps autoregressive order. The FIM corpus rewrites examples into sentinel-delimited prefix–suffix–middle order, where the spans are randomly partitioned. In particular, repeated FIM copies use different split points, so repetition is document-level exposure rather than fixed middle-span exposure. The FIM-corpus contains 50% FIM-documents for FineWeb (the rest being LTR) and 100% FIM-documents for Gutenberg.

Both models use an identical Llama 3.2 3B architecture and are trained over one epoch of $\approx$ 103B tokens ($\approx 95\%/5\%$ FineWeb/Gutenberg). Further experimental details are listed in Appendix A and model size is ablated in Section B.2.

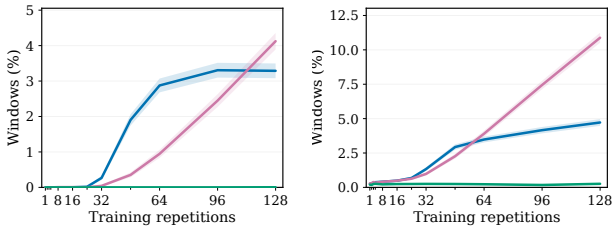
2.2. Downstream performance

We evaluate both models on 8 tasks of the LM Evaluation Harness (Gao et al., 2023), and observe that both models achieve nearly identical performance. Detailed metrics are provided in Section B.1. We conclude that differences in memorization are not due to differences in model capabilities in the context of our study.

3. Prefix-only Extraction

We compare FIM and LTR with the same prefix-only probe: using 100 prefix tokens to predict a span z of $M = 32$ target tokens. For each repetition bucket, we probe both models on the same Gutenberg windows, sampling 10 disjunct windows per excerpt.

We report two criteria. First, inspired by Cooper et al. (2026), exact extraction computes $p_z = \prod_{i=1}^{M} q_i$, where q_i is the top-k-renormalized probability of the i-th target token under $k = 40, T = 1$. A target is called *extractable* if $p_z \geq 0.1\%$. Second, we generate M tokens starting from the prefix autoregressively and report ROUGE-L (Lin, 2004), with ROUGE-L ≥ 0.5 indicating *high-overlap recovery* following Chen et al. (2025). Using $M = 32$ lets us evaluate both criteria on the same windows. This is less strict per token than the $M = 50$ setting in Cooper et al. (2026) (80.6% vs. 87.1% geometric mean). We vary M in Figure 3.



(a) Verbatim extraction rate *(b)* High-overlap recovery rate

Figure 1. **Memorization across repetition buckets.** For strict full-span extraction, LTR is higher in aggregate, but FIM extracts more windows at the largest repetition bucket. FIM yields stronger high-overlap recovery for high repetitions. FineWeb is the baseline trained only on FineWeb. Shaded bands denote nominal 95% confidence intervals for the per-window rate.

For the exact extraction criterion, LTR overall memorizes more windows: 3,279 windows satisfy $p_z \geq 0.1\%$, versus 2,230 for FIM. FIM is slightly higher on broader recovery measures, including mean ROUGE-L (0.198 for FIM vs 0.190 for LTR), and mean top-k support rate (87.09% vs 86.18%), i.e., the fraction of reference tokens contained in the top-k of logits with $k = 40$. The low memorization rate is partly due to probe position. Beginning-of-excerpt probes memorize significantly more than randomly sampled windows (Figure 7 of Section B.3).

While the FIM model's support is higher, probability mass is less concentrated on complete 32-token continuations. The exact extraction criterion is strict, such that few low-probability tokens can collapse the p_z of a target span. A threshold sweep at repetition 128 confirms this: Figure 2 shows that FIM has more mass at moderate p_z, but LTR has the heavier tail, and therefore extracts more at the 0.1% threshold.

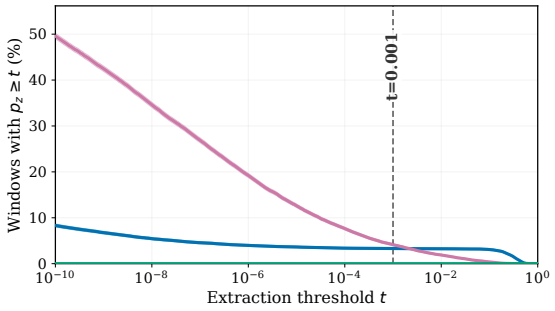


Figure 2. **Extraction survival curves** at repetition 128 show that FIM assigns more mass to moderately likely targets, but LTR has the heavier high-confidence tail. Each line gives the percentage of evaluated target windows with $p_z \geq t$ as the extraction threshold t varies. The 95% confidence intervals are smaller than the line width.

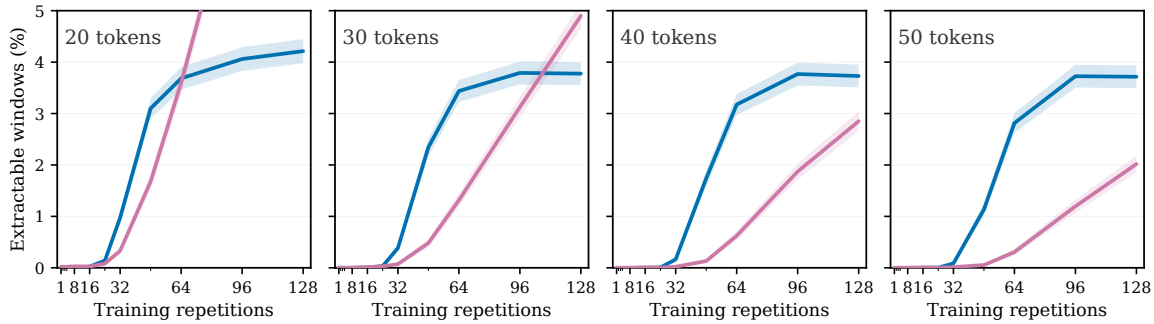


Figure 3. **Extraction rates under varying target lengths** show that the repetitions required for FIM to overtake LTR increases with span length, because longer spans favor LTR's heavier tail. Curves show the fraction of windows with $p_z \geq 0.1\%$ for the first 20, 30, 40, and 50 target tokens; all panels use the same y-axis scale. Shaded bands denote nominal 95% confidence intervals for the per-window rate.

In line with Huang et al. (2024), we find that non-trivial repetitions are required for memorization. This is expected, especially at the 3B model scale, since memorization increases with model capacity (Carlini et al., 2023). We study a 1B ablation in Section B.2. With more repetitions, LTR extraction shows diminishing returns, consistent with the logarithmic trend reported in Carlini et al. (2023). While FIM-extraction rises more steadily with repetitions, it remains low for small repetition counts. We ablate the target length in Figure 3 and conclude that the number of repetitions required for FIM to surpass LTR in extraction increases with span length. This is because a longer target makes extraction stricter, such that LTR's heavy-tailed distribution dominates.

We analyze attention patterns to further contextualize our insights. For each target-position prediction query, we partition the attention between (i) the prefix tokens and (ii) the already-seen target tokens. The latter is zero for the first target token of the target span and, for later positions, includes all earlier target tokens in the target span. We average over target positions and windows and report the mean attention allocation in Table 1. The FIM model places more attention on the prefix and less on already-seen target tokens compared to the LTR model.

Our observations can be explained by the structure of the FIM objective. Repeated LTR examples present each passage under the same left-to-right view. This concentrates probability mass into fewer long continuations, leading to the heavy-tailed distribution with increased extraction. Repeated FIM examples instead expose the same passage through varied prefix–middle–suffix decompositions, spreading mass across more partial reconstructions and broadening recoverability.

4. Native FIM probing

Since the native FIM-format includes both left and right context, it fundamentally differs from the prefix-only extraction prompt. We study the FIM-native format to evaluate how prefix and suffix context redistribute attention and contribute to memorization. As before, we sample 10 disjunct windows for each excerpt and the target remains 32 tokens. However, the 100-token context is now split across prefix and suffix. Additionally, we focus our analyses on the 128-repetition bucket, in which memorization is most prevalent. Note that this probing format includes the FIM-sentinel tokens, so even if the suffix is empty, it still differs from the prefix prompt evaluated in Section 3.

Table 1. Mean attention allocation during prediction of the target span. Both models rely primarily on the prefix, but FIM relies on it more strongly, while LTR allocates relatively more attention to earlier target tokens. Nominal 95% confidence intervals are below 10^{-4}.

Model	Prefix attention	Previous-target attention
LTR	0.604	**0.396**
FIM	**0.646**	0.354

In Figure 4, we vary the prefix–suffix split around a fixed target to test which side of the native FIM context contributes more to memorization support. As the prefix grows and the suffix shrinks, top-k support increases monotonically. The same trend holds within all repetition buckets and for both extraction rates and target likelihood (see Section B.3). In all repetition buckets, moving from suffix-only to prefix-only context, target perplexity falls from 60.23 to 27.93, while top-k support rises from 77.60% to 85.52%. The sharp drop when little or no prefix is available reflects the autoregressive structure of causal language models: without left context, the model has no reliable starting point for generating the middle span.

While prefix-heavy native FIM prompts elicit stronger memorization, the suffix still provides conditioning. The attention analysis in Figure 5 shows substantial attention allocated to both prefix and suffix, with the prefix receiving

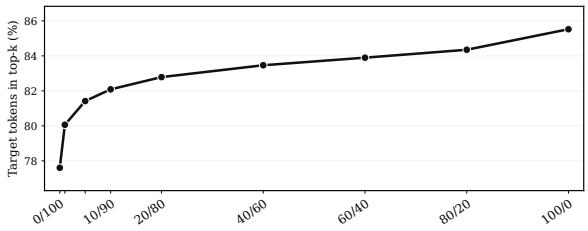


Figure 4. **Target-token top-k support** under native FIM geometry at 128 repetitions shows that memorization improves monotonically as more of the 100-token context budget is allocated to the prefix rather than the suffix. The x-axis varies `prefix`/`suffix` lengths. The line shows the percentage of target tokens included in top-40 support. The 95% confidence intervals are smaller than the line width.

slightly more attention. For prompts with very little prefix, the model compensates by attending more heavily to preceding tokens of the target span.

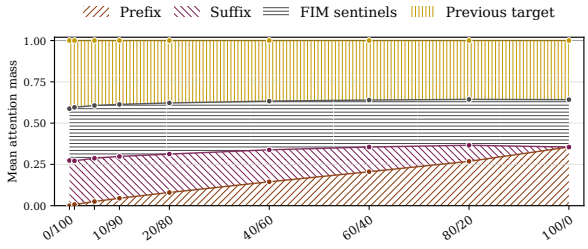


Figure 5. **Attention allocation** under native FIM probing shows that the model uses both surrounding contexts, with more attention on the prefix than the suffix, and shifts attention toward earlier target tokens when little prefix is available. The stacked areas show mean attention mass assigned to prefix tokens, suffix tokens, FIM sentinels, and earlier target tokens within the target span, averaged over target-token prediction queries and repetition buckets. The x-axis varies `prefix`/`suffix` lengths.

To isolate the contribution of prefix and suffix context directly, we keep the target fixed and replace the prefix, the suffix, or both with same-length unrelated *distractor spans* from different Gutenberg excerpts. We consider excerpts in the 128-repetition bucket and vary the prefix–suffix ratio, keeping the total context budget fixed. Figure 6 shows the top-k support in this setting. We deduce that prefix and suffix are not equally significant. Recall is strongest when the available context is allocated to the prefix. As expected, the full prompt yields the strongest top-k support across the sweep, serving as an upper-bound reference for the distractor-span conditions. While replacing the suffix with a distractor reduces recall, replacing the prefix has a significantly larger effect. When both sides are replaced by distractors, we verify that support drops sharply, confirming that the effect is not only due to prompt length or sentinel structure.

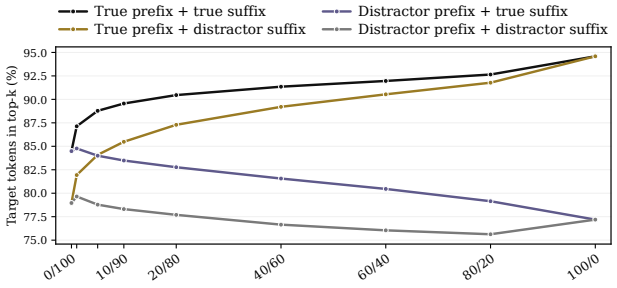


Figure 6. **Target-token top-k support** under native FIM prompting at 128 repetitions and different distractor conditions. Replacing the prefix harms recall more than replacing the suffix, confirming that prefix context is the stronger driver of memorization. The x-axis varies `prefix`/`suffix` lengths. The 95% confidence intervals are smaller than the line width.

5. Conclusion

Matched LTR and FIM models trained on a corpus containing repeated book excerpts show that the pretraining objective shapes how memorization accumulates. Under prefix-only probes, FIM improves short-span and overlap-based recovery, especially at high repetitions, while LTR produces more high-confidence long exact continuations.

Repetitions are not identical under the two objectives. In LTR, repeated excerpts reinforce the same single left-to-right view of each excerpt, and extraction grows logarithmically before saturating. In FIM, the same repeated excerpts appear in different prefix–middle–suffix decompositions. This makes memorization slower at first, but it can exceed LTR on short-span extraction at high repetition. Native FIM probes further show that while suffixes help, a short true prefix is necessary for extraction. Replacing the true prefix with a distractor prefix nearly suppresses memorization, while replacing the true suffix with a distractor suffix has a smaller effect. Our results show that LTR and FIM expose different memorization profiles and that memorization in FIM remains strongly anchored to the prefix.

5.1. Limitations and Outlook

Since we pretrain from scratch, we are not able to study frontier-scale models. Repetition counts are bounded to 128, covering a practically relevant range, but do not allow extrapolation in the limit. The main conceptual limitation is attribution: under random FIM decompositions, a probed span need not match a specific middle span seen during training, so the results do not allow us to trace exact exposures. Beyond our random-window probes, future work can investigate how prompt position impacts FIM and use span-to-training mappings to test whether the patterns persist across different probes and longer extraction windows.

Acknowledgements

We thank Yixuan Xu and Imanol Schlag for their guidance and feedback. This work was supported as part of the Swiss AI Initiative by compute grant infra01 from the Swiss National Supercomputing Centre (CSCS) on Alps.

Impact Statement

This work studies how fill-in-the-middle pretraining affects verbatim extraction of repeated text. The results can inform data curation and memorization audits for models with infilling capability. We do not introduce a new attack or release sensitive training examples. The main risk is that better measurement may also help identify settings where extraction is easier; we view this as necessary for evaluating and reducing memorization in deployed systems.

References

Bavarian, M., Jun, H., Tezak, N. A., Schulman, J., McLeavey, C., Tworek, J., and Chen, M. Efficient training of language models to fill in the middle. *ArXiv*, abs/2207.14255, 2022. URL https://api.semanticscholar.org/CorpusID:251135268.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pp. 267–284, 2019.

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

Chen, T., Brahman, F., Liu, J., Mireshghallah, N., Shi, W., Koh, P. W., Zettlemoyer, L., and Hajishirzi, H. ParaPO: Aligning language models to reduce verbatim reproduction of pre-training data. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Uic3ojVhXh.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Cooper, A. F., Lemley, M. A., Casasola, A., Ahmed, A., Gokaslan, A., Cyphert, A. B., Sa, C. D., Ho, D. E., and Liang, P. Extracting memorized pieces of (copyrighted) books from open-weight language models, 2026. URL https://arxiv.org/abs/2505.12546.

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Zhang, H., Ding, H., Xin, H., Gao, H., Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Wang, J., Chen, J., Chen, J., Yuan, J., Qiu, J., Li, J., Song, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Xu, L., Xia, L., Zhao, L., Wang, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang, M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Wang, Q., Zhu, Q., Chen, Q., Du, Q., Chen, R. J., Jin, R. L., Ge, R., Zhang, R., Pan, R., Wang, R., Xu, R., Zhang, R., Chen, R., Li, S. S., Lu, S., Zhou, S., Chen, S., Wu, S., Ye, S., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Pan, S., Wang, T., Yun, T., Pei, T., Sun, T., Xiao, W. L., Zeng, W., Zhao, W., An, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X., Chen, X., Zhang, X., Chen, X., Nie, X., Sun, X., Wang, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yu, X., Song, X., Shan, X., Zhou, X., Yang, X., Li, X., Su, X., Lin, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhu, Y. X., Zhang, Y., Xu, Y., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Yu, Y., Zheng, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Tang, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Wu, Y., Ou, Y., Zhu, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Zha, Y., Xiong, Y., Ma, Y., Yan, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Wu, Z. F., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Huang, Z.,

Zhang, Z., Xie, Z., Zhang, Z., Hao, Z., Gou, Z., Ma, Z., Yan, Z., Shao, Z., Xu, Z., Wu, Z., Zhang, Z., Li, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Gao, Z., and Pan, Z. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Fried, D., Aghajanyan, A., Lin, J., Wang, S., Wallace, E., Shi, F., Zhong, R., Yih, S., Zettlemoyer, L., and Lewis, M. Incoder: A generative model for code infilling and synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hQwb-lbM6EL.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Hayes, J., Swanberg, M., Chaudhari, H., Yona, I., Shumailov, I., Nasr, M., Choquette-Choo, C. A., Lee, K., and Cooper, A. F. Measuring memorization in language models via probabilistic extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9266–9291, 2025.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Huang, J., Yang, D., and Potts, C. Demystifying verbatim memorization in large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10711–10732, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.598. URL https://aclanthology.org/2024.emnlp-main.598/.

Kandpal, N., Wallace, E., and Raffel, C. Deduplicating training data mitigates privacy risks in language models. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pp. 10697–10707. PMLR, 2022. URL https://proceedings.mlr.press/v162/kandpal22a.html.

Kharitonov, E., Baroni, M., and Hupkes, D. How BPE affects memorization in transformers. *CoRR*, abs/2110.02782, 2021. URL https://arxiv.org/abs/2110.02782.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.577. URL https://aclanthology.org/2022.acl-long.577/.

Li, R., allal, L. B., Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., LI, J., Chim, J., Liu, Q., Zheltonozhskii, E., Zhuo, T. Y., Wang, T., Dehaene, O., Lamy-Poirier, J., Monteiro, J., Gontier, N., Yee, M.-H., Umapathi, L. K., Zhu, J., Lipkin, B., Oblokulov, M., Wang, Z., Murthy, R., Stillerman, J. T., Patel, S. S., Abulkhanov, D., Zocca, M., Dey, M., Zhang, Z., Bhattacharyya, U., Yu, W., Luccioni, S., Villegas, P., Zhdanov, F., Lee, T., Timor, N., Ding, J., Schlesinger, C. S., Schoelkopf, H., Ebert, J., Dao, T., Mishra, M., Gu, A., Anderson, C. J., Dolan-Gavitt, B., Contractor, D., Reddy, S., Fried, D., Bahdanau, D., Jernite, Y., Ferrandis, C. M., Hughes, S., Wolf, T., Guha, A., Werra, L. V., and de Vries, H. Starcoder: may the source be with you! *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=KoFOg41haE. Reproducibility Certification.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Mattern, J., Mireshghallah, F., Jin, Z., Schölkopf, B., Sachan, M., and Berg-Kirkpatrick, T. Membership in-

ference attacks against language models via neighbourhood comparison. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.719. URL https://aclanthology.org/2023.findings-acl.719/.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Nasr, M., Rando, J., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Tramèr, F., and Lee, K. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vjel3nWP2a.

Penedo, G., Kydlíček, H., allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C., Werra, L. V., and Wolf, T. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Project Gutenberg. Project gutenberg. https://www.gutenberg.org, n.d. Accessed: 2026-05-04.

Rozière, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X., Adi, Y., Liu, J., Remez, T., Rapin, J., Kozhevnikov, A., Evtimov, I., Bitton, J., Bhatt, M. P., Ferrer, C. C., Grattafiori, A., Xiong, W., D'efossez, A., Copet, J., Azhar, F., Touvron, H., Martin, L., Usunier, N., Scialom, T., and Synnaeve, G. Code llama: Open foundation models for code. *ArXiv*, abs/2308.12950, 2023. URL https://api.semanticscholar.org/CorpusID:261100919.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zWqr3MQuNs.

Shilov, I., Meeus, M., and de Montjoye, Y.-A. The mosaic memory of large language models. *Nature Communications*, 17(1), Jan 2026. ISSN 2041-1723. doi: 10.1038/s41467-026-68603-0. URL http://dx.doi.org/10.1038/s41467-026-68603-0.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1421. URL https://doi.org/10.18653/v1/n19-1421.

Xu, Y., Bosselut, A., and Schlag, I. Positional fragility in LLMs: How offset effects reshape our understanding of memorization risks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=7dBPm5c5ue.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., and Carlini, N. Counterfactual memorization in neural language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=67o9UQgTD0.

A. Experimental Details

A.1. Training Parameters

Both paired models use the Llama 3.2 3B architecture implemented in Megatron-LM with packed sequences and FlashAttention. Table 2 gives the fixed backbone configuration.

Table 2. **Backbone parameters.** The LTR and FIM runs use the same tokenizer, architecture, precision, and context-length configuration.

Parameter	Value
Layers	28
Hidden size	3072
Attention heads	24
KV heads	8
FFN hidden size	8192
Vocab size	128256
Max position embeddings	131072
RoPE base	500000
Precision	bfloat16
Dropout	0

Both paired runs use packed THD-format sequences, sequence length 16,384, micro-batch size 1, no dropout, and global batch size 2048 across 64 GH200 GPUs. One optimization step therefore consumes 33,554,432 tokens. The LTR run performs 3,057 updates over 102.58B tokens. The FIM run performs 3,064 updates over 102.81B tokens. We release training logs at `https://wandb.ai/memorization-study-fim-team/memorization-study-fim/` and model checkpoints on HuggingFace: FIM 3B, LTR 3B, FIM 1B, LTR 1B.

A.2. FIM Formatting

For each FIM document, following Bavarian et al. (2022), we randomly sample two split points within the document segment, yielding a prefix **P**, middle span **M**, and suffix **S**. The FIM condition reuses reserved Llama special-token IDs, so we do not resize the embedding table.

$$\texttt{<|fim\_prefix|>} = 128002, \qquad \texttt{<|fim\_middle|>} = 128003, \qquad \texttt{<|fim\_suffix|>} = 128005.$$

The LTR format keeps the original order:

P **M** **S** `<|eos_token|>`

The FIM format moves the middle span after its surrounding context:

`<|fim_prefix|>` **P** `<|fim_suffix|>` **S** `<|fim_middle|>` **M** `<|eos_token|>`

For FineWeb, the FIM training uses a 50% FIM / 50% LTR mixture. For Gutenberg, every 4096-token excerpt is formatted using the FIM-format. The LTR-model is only trained on LTR sequences and contains no FIM sentinels.

A.3. Gutenberg Filtering and Deduplication

We filter Project Gutenberg to obtain fixed 4096-token excerpts whose later extraction is due to controlled exposure rather than prior web familiarity. Starting from the English split of Project Gutenberg on HuggingFace[1], we strip standard Gutenberg headers, footers, licenses, and archive boilerplate. From each cleaned book, we keep characters 10,000–80,000, tokenize with the Llama 3.2 tokenizer, split into non-overlapping 4096-token windows, score each window with the FineWeb-only Llama 3.2 3B checkpoint, and keep the highest-PPL window per book. We then remove windows with PPL > 500, which were mostly indices, glossary fragments, OCR artifacts, or unusual formatting.

[1] `manu/project_gutenberg`

We deduplicate with both semantic and lexical evidence. Excerpts are embedded with `nomic-ai/nomic-embed-text-v1.5`; a pair is removed only if cosine similarity is at least 0.96 and token 5-gram Jaccard overlap is at least 0.20. For each duplicate cluster, we keep the highest-PPL excerpt. This reduces 128,003 scored windows to 33,720 final excerpts.

The final schedule has 12 repetition buckets with exposures $1, 2, 3, 4, 8, 16, 24, 32, 48, 64, 96, 128$. Each bucket contains 2,810 base excerpts, for 1,197,060 Gutenberg training documents after replication. Bucket assignment is balanced by FineWeb-checkpoint PPL; bucket means range from 36.895227 to 36.895758. The LTR-format Gutenberg corpus has 4,904,354,820 tokens, and the FIM-format Gutenberg corpus has 4,907,946,000 tokens, with the difference coming from FIM sentinel tokens.

B. Additional Experimental Results

B.1. Downstream Performance

We report the detailed metrics of our matched Llama 3.2 3B models for the LM Evaluation Harness suite (Gao et al., 2023) in Table 3. The scores of the 1B-scale ablation in Section B.2 are also reported.

Table 3. **Downstream quality-control suite by model scale.** Accuracy tasks are reported in %; higher is better. Lower is better for Wikitext word perplexity. Within each scale, green marks the better of LTR and FIM, and red marks the worse. Bold marks the best score in the row. Δ is FIM minus LTR within each scale (pp for accuracy, absolute for PPL).

		1B			3B		
Task	Metric	LTR	FIM	Δ	LTR	FIM	Δ
MMLU (Hendrycks et al., 2021)	acc % ↑	23.57	22.89	-0.68	23.81	**24.42**	+0.61
HellaSwag (Zellers et al., 2019)	acc % ↑	32.40	32.65	+0.25	36.36	**36.95**	+0.60
ARC-Challenge (Clark et al., 2018)	acc % ↑	19.28	19.28	+0.00	**22.18**	21.25	-0.94
ARC-Easy (Clark et al., 2018)	acc % ↑	48.36	47.52	-0.84	**53.24**	52.53	-0.72
CommonsenseQA (Talmor et al., 2019)	acc % ↑	19.57	19.57	+0.00	**19.98**	19.74	-0.25
PIQA (Bisk et al., 2020)	acc % ↑	67.90	67.03	-0.87	69.80	**70.40**	+0.60
WinoGrande (Sakaguchi et al., 2020)	acc % ↑	**51.22**	51.07	-0.16	51.14	50.43	-0.71
Wikitext (Merity et al., 2017)	PPL ↓	26.19	26.30	+0.11	**25.25**	26.98	+1.73

B.2. Model size ablation

Memorization has been shown to increase with model capacity (Carlini et al., 2023). To test the validity and generalizability of our conclusions, we train paired Llama 3.2 1B models in the same conditions.

We observe that, as expected, both downstream performance and verbatim memorization decrease at smaller scale. Table 3 reports the downstream comparison, and Figure 7 shows reduced memorization relative to the 3B variant. Because exact extraction on the random-window probes used in Section 3 is too rare at 1B scale for a stable comparison, we focus on ROUGE-L instead.

Importantly, note that the relative trends between LTR and FIM remain consistent with our main results in Section 3.

B.3. Additional Figures

Figures 7 to 9 show additional figures omitted from the main text.

B.4. Qualitative Assessment of Memorization

Figures 10 to 12 show examples of memorized windows that are extractable by both models (Figure 10), only extractable by the FIM-model (Figure 11), and only extractable by the LTR-model (Figure 12).

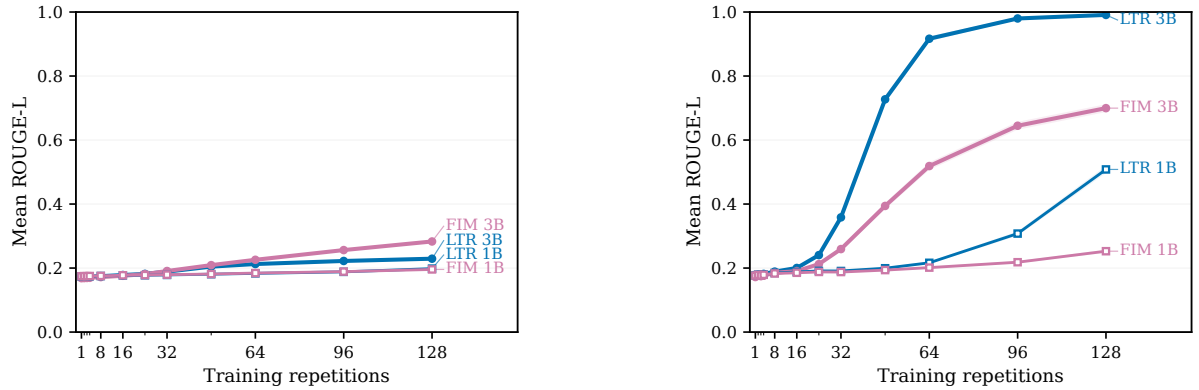
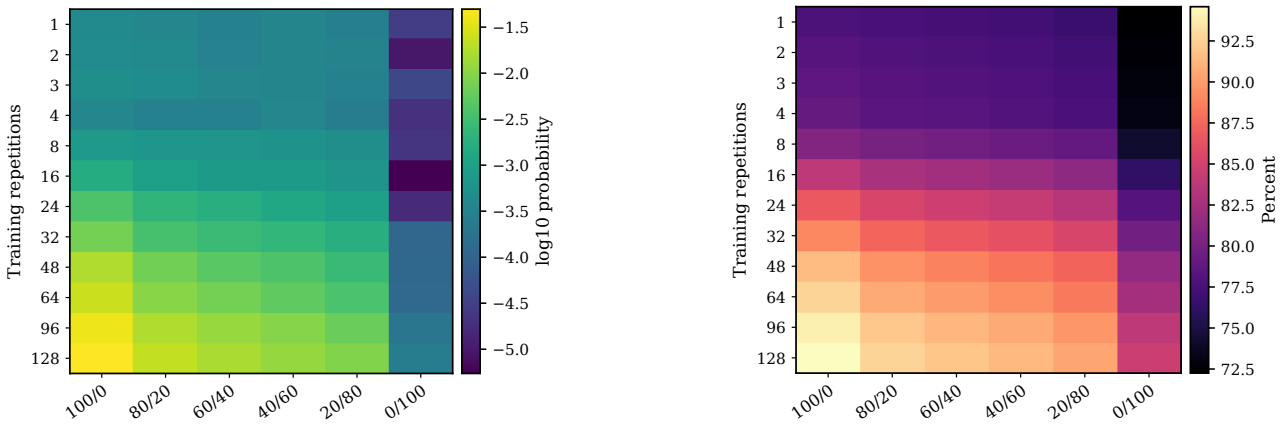


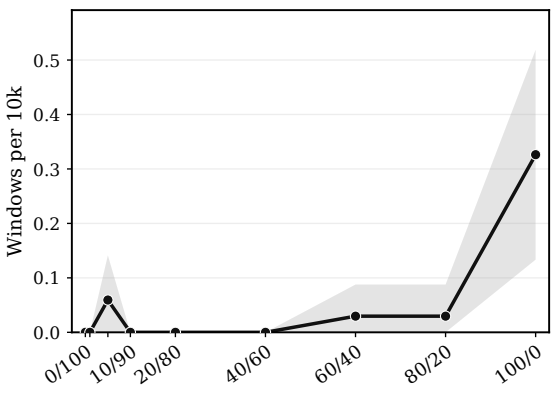
Figure 7. **Mean ROUGE-L under prefix probing for 1B and 3B models**, evaluated on 10 uniformly sampled windows per excerpt (left) and on the first window of each excerpt (right). Each prompt uses 100 prefix tokens to generate a 32-token continuation. Filled circles denote 3B models; hollow squares denote 1B models. The large gap between first-window and uniformly sampled-window probing indicates that recall is anchored near the beginning of repeated excerpts, consistent with positional fragility observed by Xu et al. (2026).



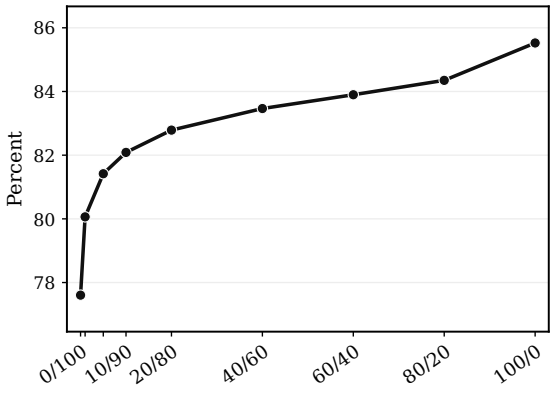
(a) Mean per-token Cooper probability.

(b) Target tokens in top-k.

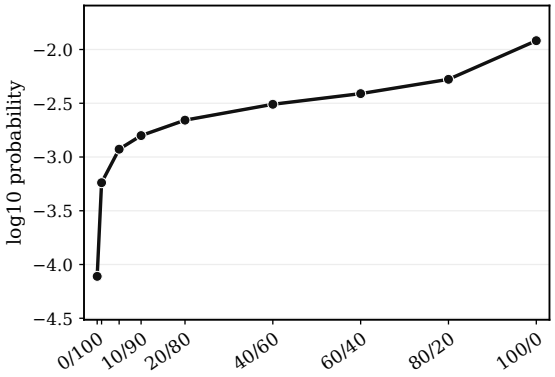
Figure 8. **Native FIM geometry by repetition bucket.** Heatmaps separate the prefix–suffix effect across repetition levels. The x-axis varies `prefix`/`suffix` lengths.



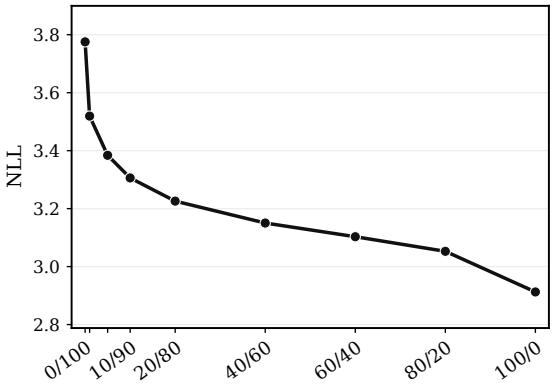
(a) Extractability



(b) Target tokens in top-k

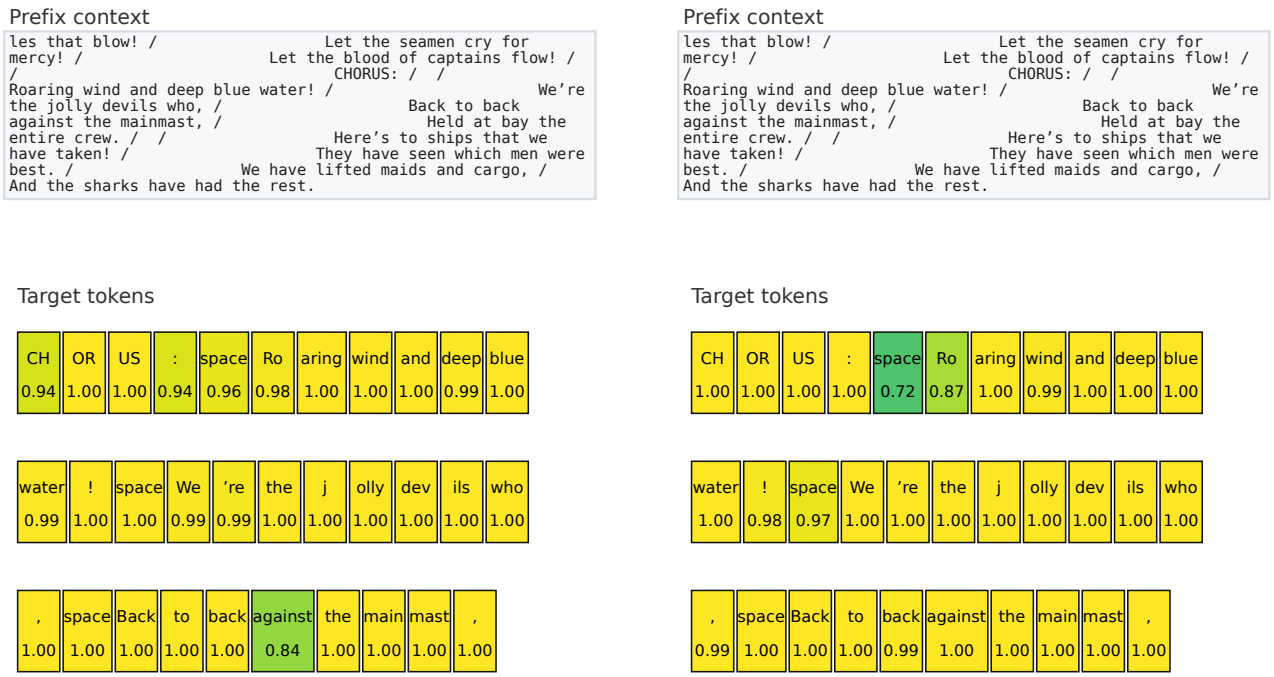


(c) Mean per-token top-k renormalized log-probability.



(d) Teacher-forced target NLL

Figure 9. **Native FIM probing across prefix–suffix geometry.** Metrics are over *all* repetition buckets. The x-axis varies `prefix`/`suffix` lengths. Shaded bands are nominal 95% confidence intervals.

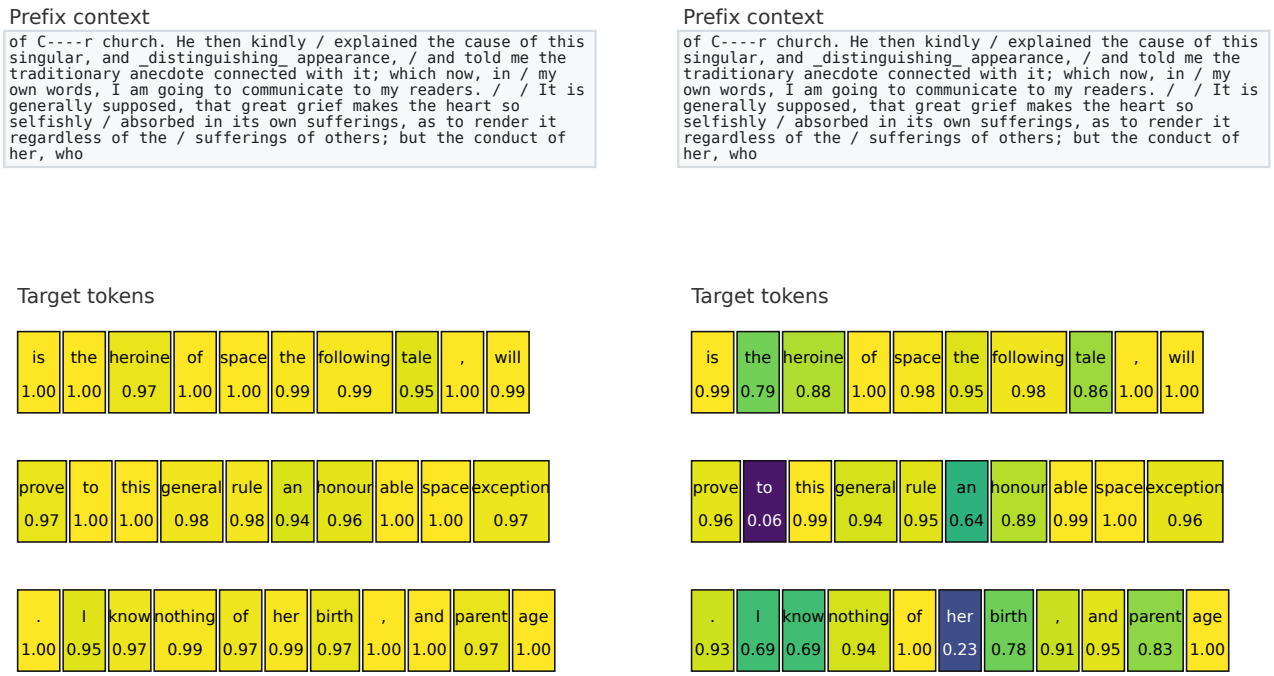


(a) LTR *(b)* FIM

Figure 10. Window that was extracted by both models. Numbers indicate the top-k re-normalized logits of the displayed true target tokens. Repetition 128; source book 54068-0; excerpt 54068-0::window_0000; target start 100; prefix length 100 tokens; target length 32 tokens; p_z values: LTR=0.711046, FIM=0.585069.

(a) LTR *(b)* FIM

Figure 11. Window that was only extracted by the FIM-model. Numbers indicate the top-k re-normalized logits of the displayed true target tokens. Repetition 128; source book 57335-0; excerpt 57335-0::window_0002; target start 100; prefix length 100 tokens; target length 32 tokens; p_z values: LTR=0.000219776, FIM=0.204912.



(a) LTR *(b)* FIM

Figure 12. Window that was only extracted by the LTR-model. Numbers indicate the top-k re-normalized logits of the displayed true target tokens. Repetition 128; source book 11326-8; excerpt 11326-8::window_0003; target start 100; prefix length 100 tokens; target length 32 tokens; p_z values: LTR=0.588202, FIM=0.00063823.