# Internal Data Repetition Destroys Language Models

**author names withheld**

**Under Review for the Workshop on High-dimensional Learning Dynamics, 2026**

## Abstract

Language models are running out of high-quality training data, and even aggressively deduplicated corpora retain some amount of repetition. Earlier controlled studies predated Chinchilla-style scaling laws and could only measure the cost of repetition indirectly. We revisit repetition in the Chinchilla-style scaling regime, using a fitted no-repetition scaling law to report Compute-Equivalent Gain and Compute-Equivalent Loss. We show that repetition damage in this modernized regime is systematic in three ways. First, holding compute allocated to repeated data constant, eval loss peaks at an intermediate repeat count $R$; repeating a moderately sized subset a moderate number of times damages performance more than repeating a large subset a few times or a small subset many times. Second, the location of this peak is well-fit by a power law in model size. Finally, when repeated documents consume 10% of the FLOPS budget in a controlled exact-document repetition setting, the compute-equivalent loss can be large: on FineWeb-Edu-Dedup, the most damaging repeat count for a Qwen3-style 344M-parameter model at $\mathrm{OT} = 1$ matches the loss of a no-repetition run using about 67% of the FLOPs. We demonstrate that these phenomena are not language-model-specific, and can be analytically understood in a simple statistical model: a misspecified linear regression with verbatim duplicates reproduces the same qualitative loss peak, suggesting that such peaks can arise from a statistical tradeoff between memorization and generalization. Our findings add precision to the study of duplication in language models, allowing practitioners to quantify the wasted compute incurred by the presence of duplicates.

## 1. Introduction

Pretraining has entered a data-constrained regime. The high-quality public text corpora used for frontier training are within a small constant factor of being exhausted [33, 55], and recent flagship corpora such as FineWeb-Edu, DataComp-LM, Dolma, and RedPajama-v2 [32, 44, 48, 56] have responded with more aggressive deduplication and filtering. Yet aggressive deduplication is not perfect deduplication [1, 30, 53]. Pretraining streams continue to contain near-duplicate documents, paraphrased templates, and semantically redundant web pages, and as scale grows the meaning of "duplicate" itself shifts [27]. Replication and duplicates can vary in type and effect. In this paper, we isolate one controlled case: exact document-level replay of a selected repeated pool. Our question is how much this controlled form can cost.

The closest earlier controlled study, Hernandez et al. [22], established that repeating a small fraction of training data degrades held-out loss in a non-monotonic way, and framed the damage as a reduction in *effective parameter count*. That framing was natural before Chinchilla-style scaling [23] provided a clean compute axis for predictions. The quantity a practitioner allocating a pretraining budget cares about is how many FLOPs a no-repetition run would need to reach the same loss. We replace effective parameter count with *Compute-Equivalent Gain (*$\mathrm{CEG}$*)*, following

the compute-equivalent gains framework of Davidson et al. [15], Gundlach et al. [19], Meta Superintelligence Labs [39]. For a repeated-data run with loss $L$ and actual compute $C_{\text{actual}}$, CEG is the no-repetition compute required to reach $L$ divided by $C_{\text{actual}}$. We define *Compute-Equivalent Loss* as $\text{CEL} = 1 - \text{CEG}$. Thus $\text{CEG} = 1$ matches the no-repetition reference, while $\text{CEG} < 1$ indicates compute-equivalent loss.

First, we measure repeated-data damage in compute-equivalent units using a fitted no-repetition scaling law. Second, we show that eval loss peaks at an intermediate repeat count and that the peak location follows a power-law trend in model size. Third, in Appendix A, we give a misspecified linear-regression analogue that reproduces the same qualitative non-monotonicity.

We defer Related Work to Appendix B and Appendix C.

## 2. Methods

Repeated data is increasingly hard to avoid, so the practical question becomes which repeat structures are dangerous and how much compute they waste. We fix the repeated-token fraction at $f = 0.1$ and vary only its concentration: the repeated 10% can come from a large pool of data seen a few times or a small pool seen many times. This fraction is large enough to produce measurable damage while keeping the bulk of training on unique data, and matches the setup used by Hernandez et al. [22]. It also isolates repetition *structure* from repetition *amount*, letting us identify the most harmful configurations at fixed compute.

**Repeated-pool construction.** For each repeated-data run, $(1 - f)T$ tokens are drawn from non-repeated documents. Let $D_{\text{r}}$ denote the number of unique tokens in the repeated pool, and let $R$ denote the number of times each repeated document is replayed. The repeated-token budget is therefore $fT \approx RD_{\text{r}}$ so

$$D_{\text{r}} \approx \frac{fT}{R} = \frac{2 \cdot \text{OT} \cdot N}{R}. \tag{1}$$

Increasing $R$ does not change the amount of repeated material we train on; it concentrates the same 10% repeated-token budget onto a smaller pool. The $R$ view answers how many times repeated documents are replayed, while the $D_{\text{r}}$ view answers how large the repeated corpus is. Both are needed because repetition damage depends on their interaction.

The repeated documents are sampled at document granularity and are disjoint from the non-repeated training documents. Copies of repeated documents are randomly interleaved with the non-repeated stream during training. This setup gives us a controlled setting to study repetition—which is unavoidable in any realistic training corpus—and identify the configurations that cause the most damage. See Appendix F for more details on the sampling protocol.

**Evaluation and baselines.** We evaluate every model on a fixed held-out split of approximately 150M tokens, constructed with a fixed train/test seed. This evaluation split is excluded from all training data and from all repeated pools by the fixed train/test split. For each completed $(N, \text{OT})$ sweep, we also train a no-repetition baseline. These baselines serve two purposes. First, the baseline at the same $(N, \text{OT})$ gives the per-sweep reference for measuring the fractional eval-loss increase caused by repetition. Second, the six $\text{OT} = 1$ no-repetition baselines calibrate the no-repetition Chinchilla scaling law $L(C) = E + KC^{-\gamma}$ used in §3.3 to convert eval loss into CEG and CEL. Additional implementation and evaluation details are given in Appendix D and Appendix H.

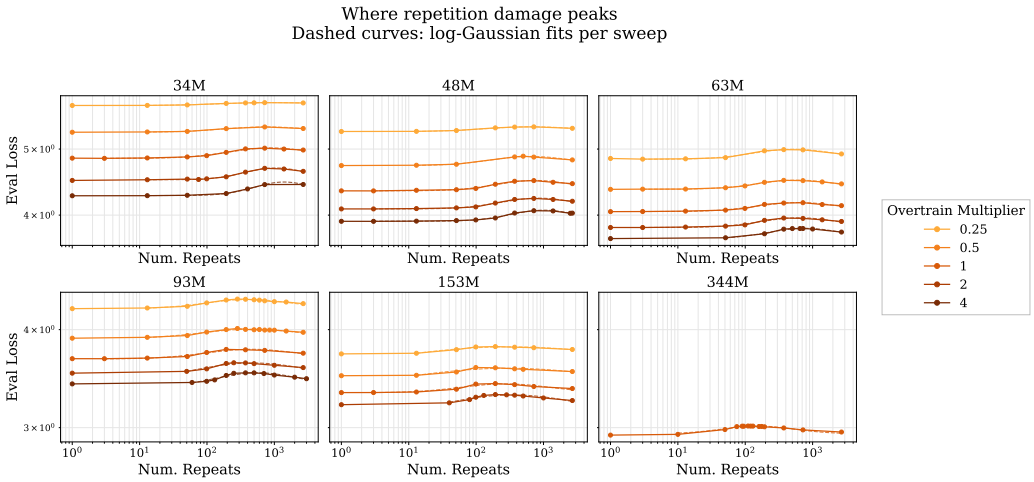


Figure 1: Gaussian fits to eval loss as a function of repeat count. Each panel fixes a model size $N$, and each curve corresponds to an overtraining multiplier OT. The fitted peak gives $R^{\text{peak}}$, the repeat count at which eval loss is largest for that $(N, \text{OT})$ sweep. Across all model sizes, eval loss is maximized at an intermediate repeat count.

## 3. Results

We characterize how eval loss depends on the structure of the repeated subset (§3.1), extract a model-size scaling law for the worst-case configuration (§3.2), and translate the resulting loss differences into CEG and CEL via the no-repetition Chinchilla scaling law (§3.3). In Appendix A–A.1, we give a closed-form linear-regression analogue that reproduces the same qualitative non-monotonicity.

### 3.1. Eval loss is non-monotonic in the repeat count

For each $(N, \text{OT})$ pair we hold the repeated fraction $f = 0.1$ and the total compute $C$ fixed, varying only $R$ along an iso-FLOP curve. Figure 1 plots eval loss against $R$. The qualitative pattern is an intermediate-repeat peak with a sharpness that varies between sweeps. Across the completed sweeps, the raw maximum is 1.0 to 4.2% above the corresponding no-repetition baseline, with median 3.1%. Measured relative to the larger of the two endpoints, no repeats and the largest measured $R$, the peak prominence ranges from 0.7 to 2.7%, with median 1.8%. The peaks are often broad. We therefore use the log-Gaussian fits in §3.2 to estimate peak locations, rather than treating the discrete argmax as exact, because $R$ is sampled on a finite, approximately logarithmic grid and the true maximum may fall between measured repeat counts. Hernandez et al. [22] reported a similar non-monotonic dependence in a fixed 100B-token-budget setting. We confirm that the same qualitative intermediate-repeat regime appears under Chinchilla-style budgets, and quantify it using CEG and CEL in §3.3. The implication is that the most damaging repetition structure usually lies away from both extremes of the $R$ range. Configurations with many repeats of a tiny pool, or few repeats of a large pool, generally produce smaller loss increases than configurations in between,

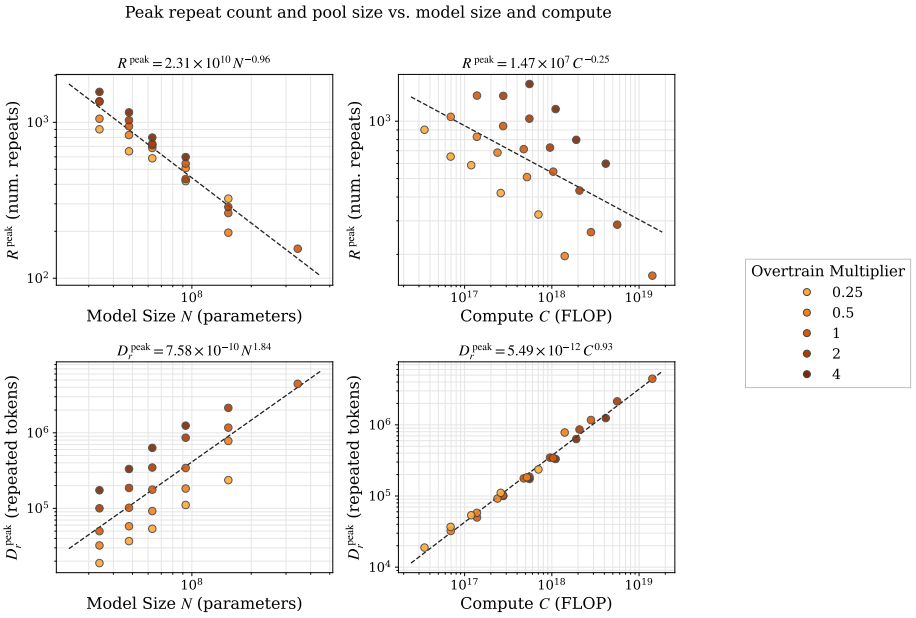


Figure 2: These fits predict the most damaging repetition structure for a given training budget. The repeat count at peak eval loss $R^{\text{peak}}$ decreases with model size and compute. The repeated-pool size at peak eval loss $D_{\text{r}}^{\text{peak}}$ increases with both.

though the recovery at large $R$ is flatter in some sweeps. Appendix A offers a statistical mechanism for why such a peak can arise.

### 3.2. Peak locations are consistent with a power-law trend in model size

§3.1 established that loss is maximized at some intermediate $R$. To summarize how this peak shifts with scale, we fit a simple empirical relationship between the estimated peak location and model size. We extract continuous peak locations by fitting a log-Gaussian to eval loss as a function of $\log_{10} R$ for each $(N, \text{OT})$ pair,

$$L(x) = b + A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad R^{\text{peak}} = 10^{\mu}. \tag{2}$$

This fit captures the single peak we observe in log-repeat space. We then fit power laws to the estimated $R^{\text{peak}}$ values across the completed $(N, \text{OT})$ grid and convert them to repeated-pool sizes using (1), giving:

$$R^{\text{peak}} = 2.31 \times 10^{10} N^{-0.96}, \qquad D_{\text{r}}^{\text{peak}} = 7.58 \times 10^{-10} N^{1.84}, \tag{3}$$

$$R^{\text{peak}} = 1.47 \times 10^{7} C^{-0.25}, \qquad D_{\text{r}}^{\text{peak}} = 5.49 \times 10^{-12} C^{0.93}. \tag{4}$$

Figure 2 plots these four related views. Because the peak locations are themselves estimated from fitted curves, and because the model sizes span roughly one order of magnitude within a single

architecture family, we interpret these fits as within-range empirical summaries rather than validated extrapolative scaling laws.

**Observed trend.** Within the measured range, larger models tend to reach their largest loss increase at fewer repeats of a larger repeated pool. The Qwen3-style 34M-parameter model peaks at $R \approx 1400$ with $D_{\mathrm{r}} \approx 5 \times 10^4$ tokens, while the Qwen3-style 344M-parameter model peaks at $R \approx 155$ with $D_{\mathrm{r}} \approx 4.5 \times 10^6$ tokens [60]. We also do not observe a strong dependence on training duration over the completed grid: the OT $\in \{0.25, 0.5, 1, 2, 4\}$ sweeps largely fall near the same trend in $N$. Thus, the fitted trend provides a useful heuristic for identifying repetition regimes that may be especially harmful within the studied scale range.

### 3.3. Repetition can cause an $\mathcal{O}(1)$ Compute-Equivalent Loss

The 2 to 4% loss increases reported in §3.1 and §3.2 translate into different compute gaps depending on where the run sits on the no-repetition Chinchilla curve. Hernandez et al. [22] addressed this by reporting damage as a reduction in effective parameter count, but that framing predates Chinchilla-style scaling and is hard to compare across model sizes and training durations. We replace it with a compute-equivalent ratio. We ask how much compute a no-repetition run would need to match the loss of a repeated-data run. We fit the no-repetition Chinchilla scaling law $L(C) = E + KC^{-\gamma}$ to the six OT=1, $R$=1 baselines [17, 23, 45], obtaining $L(C) = 2.365 + 6.647 \times 10^5 C^{-0.317}$. The fit is shown in Figure 6. Here $E$ is the irreducible-loss floor that the model approaches in the limit of infinite compute, $K$ sets the overall vertical scale, and $\gamma > 0$ is the rate at which loss decreases with compute. Inverting the law gives $C^\star(L) = (K/(L-E))^{1/\gamma}$, the amount of compute a no-repetition run on the law would need to reach a measured loss $L$. The *Compute-Equivalent Gain* of a repeated-data run is $\mathrm{CEG} = \frac{C^\star(L)}{C_{\mathrm{actual}}}$, where $\mathrm{CEL} = 1 - \mathrm{CEG}$. $\mathrm{CEG} = 1$ matches the no-repetition law. When $\mathrm{CEG} < 1$, the run reaches the loss of a no-repetition run trained with only $\mathrm{CEG} \cdot C_{\mathrm{actual}}$ FLOPs, and $\mathrm{CEL} = 1 - \mathrm{CEG}$ is the Compute-Equivalent Loss.

Figure 8 shows CEG as a function of $R$. At OT=1, the worst repeat settings increase CEL to $0.19, 0.19, 0.21, 0.21, 0.26, 0.33$ for $N \in \{34, 48, 63, 93, 153, 344\}$M respectively. **At the largest scale, peak repetition produces CEL $\approx$ 0.33.** Two patterns are notable. First, the 2 to 4% loss bump translates into CEL values of roughly 0.19 to 0.33 because the no-repetition scaling law is shallow. A small loss gap maps to a large compute gap, and the loss-space view systematically understates the practical cost. Second, varying OT shifts the level of CEG but leaves the location of the peak in $R$ approximately unchanged. The worst-case repetition structure persists across training durations and is described in more detail in Appendix J.

## 4. Conclusion

Pretraining is now data-constrained, and this increases risk of residual repetition. We studied exact document-level repetition at fixed repeated-token fraction and fixed compute. Eval loss is maximized at an intermediate repeat count, and the peak shifts systematically with model size: larger models are most damaged by fewer repeats of larger repeated pools. Under our fitted no-repetition scaling law, the most damaging repeat setting reaches CEG $\approx 0.67$, corresponding to CEL $\approx 0.33$. These results emphasize that the structure of repetition matters, not only the total fraction of duplicated tokens.

## References


[1] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. SemDeDup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023. URL https://arxiv.org/abs/2303.09540.

[2] Madhu S. Advani, Andrew M. Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020. doi: 10.1016/j.neunet.2020.08.022. URL https://arxiv.org/abs/1710.03667.

[3] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=XfHWcNTSHp. Survey Certification, Featured Certification.

[4] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=FxNNiUgtfa.

[5] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024. doi: 10.1073/pnas.2311878121. URL https://arxiv.org/abs/2102.06701.

[6] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020. doi: 10.1073/pnas.1907378117. URL https://arxiv.org/abs/1906.11300.

[7] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. doi: 10.1073/pnas.1903070116. URL https://arxiv.org/abs/1812.11118.

[8] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020. doi: 10.1137/20M1336072. URL https://arxiv.org/abs/1903.07571.

[9] Tamay Besiroglu, Ege Erdil, Matthew Barnett, and Josh You. Chinchilla scaling: A replication attempt. *arXiv preprint arXiv:2404.10102*, 2024. URL https://arxiv.org/abs/2404.10102.

[10] Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/59404fb89d6194641c69ae99ecdf8f6d-Abstract-Conference.html.

[11] Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. A dynamical model of neural scaling laws. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 4345–4382, 2024. URL https://proceedings.mlr.press/v235/bordelon24a.html.

[12] Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sckjveqlCZ.

[13] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

[14] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

[15] Tom Davidson, Jean-Stanislas Denain, Pablo Villalobos, and Guillem Bas. AI capabilities can be significantly improved without expensive retraining, 2023. URL https://arxiv.org/abs/2312.07413.

[16] Chunyuan Deng, Yilun Zhao, Yuzhao Heng, Yitong Li, Jiannan Cao, Xiangru Tang, and Arman Cohan. Unveiling the spectrum of data contamination in language models: A survey from detection to remediation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16078–16092, 2024. URL https://aclanthology.org/2024.findings-acl.951/.

[17] Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Luca Soldaini, Jenia Jitsev, Alex Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. Language models scale reliably with over-training and on downstream tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iZeQBqJamf.

[18] Sachin Goyal, Pratyush Maini, Zachary C. Lipton, Aditi Raghunathan, and J. Zico Kolter. Scaling laws for data filtering – data curation cannot be compute agnostic. *arXiv preprint arXiv:2404.07177*, 2024. URL https://arxiv.org/abs/2404.07177.

[19] Hans Gundlach, Alex Fogelson, Jayson Lynch, Ana Trisovic, Jonathan Rosenfeld, Anmol Sandhu, and Neil Thompson. On the origin of algorithmic progress in ai, 2025. URL https://arxiv.org/abs/2511.21622.

[20] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50

(2):949–986, 2022. doi: 10.1214/21-AOS2133. URL https://projecteuclid.org/journals/annals-of-statistics/volume-50/issue-2/Surprises-in-high-dimensional-ridgeless-least-squares-interpolation/10.1214/21-AOS2133.full.

[21] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. URL https://arxiv.org/abs/2010.14701.

[22] Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022. URL https://arxiv.org/abs/2205.10487.

[23] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract-Conference.html.

[24] Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5075–5084, 2023. URL https://aclanthology.org/2023.emnlp-main.308/.

[25] Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10697–10707, 2022. URL https://proceedings.mlr.press/v162/kandpal22a.html.

[26] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

[27] Joshua Kazdan, Noam Levi, Rylan Schaeffer, Jessica Chudnovsky, Abhay Puri, Bo He, Mehmet Donmez, Sanmi Koyejo, and David Donoho. Scale dependent data duplication. *arXiv preprint arXiv:2603.06603*, 2026. URL https://arxiv.org/abs/2603.06603.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[29] Aran Komatsuzaki. One epoch is all you need. *arXiv preprint arXiv:1906.06669*, 2019. URL https://arxiv.org/abs/1906.06669.

[30] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022. doi: 10.18653/v1/2022.acl-long.577. URL https://aclanthology.org/2022.acl-long.577/.

[31] Pietro Lesci, Clara Meister, Thomas Hofmann, Andreas Vlachos, and Tiago Pimentel. Causal estimation of memorisation profiles. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 15616–15635, 2024. URL https://aclanthology.org/2024.acl-long.834/.

[32] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. doi: 10.52202/079017-0455. URL https://papers.nips.cc/paper_files/paper/2024/hash/19e4ea30dded58259665db375885e412-Abstract-Datasets_and_Benchmarks_Track.html.

[33] Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, and toxicity. In *Proceedings of the 2024 Conference of NAACL: Human Language Technologies*, pages 3245–3276, 2024. URL https://aclanthology.org/2024.naacl-long.179/.

[34] Justin Lovelace, Christian Belardi, Srivatsa Kundurthy, Shriya Sudhakar, and Kilian Q. Weinberger. Prescriptive scaling laws for data constrained training. *arXiv preprint arXiv:2605.01640*, 2026. URL https://arxiv.org/abs/2605.01640.

[35] Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 2: Short Papers*, pages 157–165, 2022. URL https://aclanthology.org/2022.acl-short.18/.

[36] Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. *arXiv preprint arXiv:2401.16380*, 2024. URL https://arxiv.org/abs/2401.16380.

[37] Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023. URL https://arxiv.org/abs/2309.04564.

[38] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022. doi: 10.1002/cpa.22008. URL https://arxiv.org/abs/1908.05355.

[39] Meta Superintelligence Labs. Introducing Muse Spark: Scaling towards personal superintelligence, April 2026. URL https://ai.meta.com/blog/introducing-muse-spark-msl/. Accessed: 2026-05-06.

[40] Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=j5BuTrEj35.

[41] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=B1g5sA4twr.

[42] Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=KS8mIvetg2.

[43] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb dataset for Falcon LLM: Outperforming curated corpora with web data only. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. URL https://papers.nips.cc/paper/2023/hash/fa3ed726cc5073b9c31e3e49a807789c-Abstract-Datasets_and_Benchmarks.html.

[44] Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. doi: 10.52202/079017-0970. URL https://papers.nips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

[45] Tomer Porian, Mitchell Wortsman, Jenia Jitsev, Ludwig Schmidt, and Yair Carmon. Resolving discrepancies in compute-optimal scaling of language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=4fSSqpk1sM.

[46] Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=0bmXrtTDUu.

[47] Rylan Schaeffer, Joshua Kazdan, Baber Abbasi, Ken Ziyu Liu, Brando Miranda, Ahmed Ahmed, Fazl Berez, Abhay Puri, Stella Biderman, Niloofar Mireshghallah, and Sanmi Koyejo. Quantifying the effect of test set contamination on generative evaluations. *arXiv preprint arXiv:2601.04301*, 2026. URL https://arxiv.org/abs/2601.04301.

[48] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: An open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 15725–15788, 2024. URL https://aclanthology.org/2024.acl-long.840/.

[49] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/7b75da9b61eda40fa35453ee5d077df6-Abstract-Conference.html.

[50] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

[51] Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Won Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Q. Tran, Dani Yogatama, and Donald Metzler. Scaling laws vs model architectures: How does inductive bias influence scaling? *arXiv preprint arXiv:2207.10551*, 2022. URL https://arxiv.org/abs/2207.10551.

[52] Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 38274–38290, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/fa0509f4dab6807e2cb465715bf2d249-Abstract-Conference.html.

[53] Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. D4: Improving llm pretraining via document de-duplication and diversification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/a8f8cbd7f7a5fb2c837e578c75e5b615-Abstract-Datasets_and_Benchmarks.html.

[54] Alexander Tsigler and Peter L. Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023. URL https://jmlr.org/papers/v24/22-1398.html.

[55] Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of LLM scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ViZcgDQjyG.

[56] Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. RedPajama: An open dataset for training large language models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. doi: 10.52202/079017-3697. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d34497330b1fd6530f7afd86d0df9f76-Abstract-Datasets_and_Benchmarks_Track.html.

[57] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/dcba6be91359358c2355cd920da3fcbd-Abstract-Conference.html.

[58] Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. To repeat or not to repeat: Insights from scaling llm under token-crisis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/b9e472cd579c83e2f6aa3459f46aac28-Abstract-Conference.html.

[59] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL https://arxiv.org/abs/2407.10671.

[60] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang,

Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

[61] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

## Appendix A. A statistical model of repetition damage

The non-monotonic peak in eval loss could plausibly be a transformer-specific artifact, a memorization quirk arising from attention, depth, or optimization dynamics. We suggest that it may reflect a more general statistical effect. A finite-capacity learner trained on duplicated samples from a richer distribution can trade fit on the repeated set against generalization beyond the repeated samples, and this tradeoff can produce non-monotonic behavior in the duplication count. We illustrate this possibility in misspecified linear regression, deriving closed-form conditional risks and recovering the same qualitative peak in simulations.

**Setup.** The data-generating process is a high-dimensional linear model with isotropic Gaussian inputs. Inputs are $x \sim \mathcal{N}(0, I_p)$ in $\mathbb{R}^p$, with noiseless labels $y = x^\top \beta$ and a fixed coefficient vector $\beta \in \mathbb{R}^p$. The learner observes only the first $m < p$ coordinates of $x$, so the model is misspecified. Decompose $x = (x_{\text{in}}, x_{\text{out}})$ and $\beta = (\beta_{\text{in}}, \beta_{\text{out}})$, with $x_{\text{in}}, \beta_{\text{in}} \in \mathbb{R}^m$ observed and $x_{\text{out}}, \beta_{\text{out}} \in \mathbb{R}^{p-m}$ unobserved. The same isotropic distribution governs the test point. The unobserved coordinates carry real predictive signal, so finite-sample correlations between $x_{\text{in}}$ and $x_{\text{out}}$ can be mistaken for useful structure. The training set contains $n$ unique examples plus a block of $d$ examples each repeated $r$ times, for a total of $n + rd$ rows.

**Block-diagonal noise covariance.** Let $X_{u,\text{in}}$ and $X_{d,\text{in}}$ stack the observed features for the unique and repeated examples respectively, and let $C_u = X_{u,\text{in}}^\top X_{u,\text{in}}$ and $C_d = X_{d,\text{in}}^\top X_{d,\text{in}}$ be the corresponding observed-feature Gram matrices. The expanded observed-feature Gram of the full training set is $B_r = C_u + rC_d$. The restricted OLS estimator decomposes as $\hat{\beta}_{\text{in}} = \beta_{\text{in}} + a_r$, where $a_r$ is an aliasing term that arises from fitting the unobserved signal through the observed coordinates. The crucial step is that the $r$ copies of each repeated example share a single unobserved-feature realization, so the conditional covariance of $z = X_{\text{out}}\beta_{\text{out}}$ given $X_{\text{in}}$ is the block-diagonal direct sum

$$\Sigma_r \;=\; I_n \;\oplus\; \bigoplus_{i=1}^{d} \mathbf{1}_r \mathbf{1}_r^\top \;\in\; \mathbb{R}^{(n+rd)\times(n+rd)}, \tag{5}$$

where $\mathbf{1}_r \in \mathbb{R}^r$ is the all-ones vector and the symbol $\oplus$ denotes block-diagonal direct sum. The unique block is the $n \times n$ identity, reflecting independent unobserved features across unique examples. Each repeated block is the rank-one $r \times r$ matrix $\mathbf{1}_r \mathbf{1}_r^\top$, reflecting that the $r$ copies of document $i$ share a single $x_{\text{out},i}$. Distinct repeated documents are uncorrelated. Multiplying through gives $X_{\text{in}}^\top \Sigma_r X_{\text{in}} = C_u + r^2 C_d$. The extra factor of $r$ is what makes duplication qualitatively different from adding more independent samples. Let

$$A_r = C_u + rC_d, \qquad G_r = C_u + r^2 C_d.$$

Then, conditioning on $X_{\text{in}}$, the expected train and test errors are

$$\mathbb{E}[L_{\text{train}} \mid X_{\text{in}}] = \frac{\|\beta_{\text{out}}\|_2^2}{n + rd} \left[ n + rd - \operatorname{tr}\left(G_r A_r^{-1}\right) \right], \tag{6}$$

$$\mathbb{E}[L_{\text{test}} \mid X_{\text{in}}] = \|\beta_{\text{out}}\|_2^2 \left[ 1 + \operatorname{tr}\left(A_r^{-1} G_r A_r^{-1}\right) \right]. \tag{7}$$

Equations (6)–(7) are closed-form conditional risks. The non-monotonic behavior is a simulation-supported mechanism explored in Appendix A.1; the full derivation is given in Appendix G.

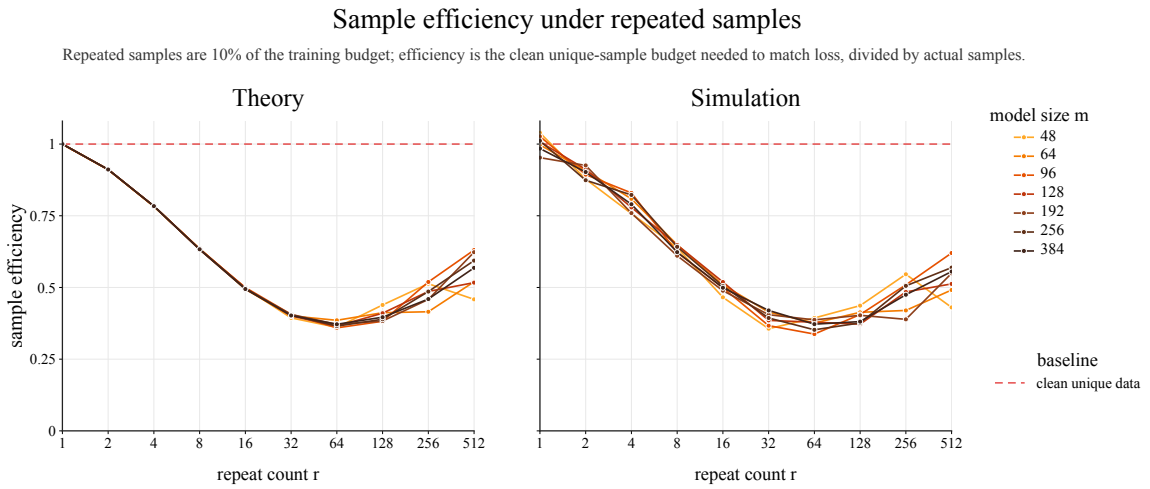


Figure 3: Sample efficiency under repeated samples. The repeated block accounts for $10\%$ of the training budget. For each repeat count $r$, SE is the unique-sample budget needed to match the repeated-data test loss, divided by the actual sample budget. Theory and simulation both show non-monotonic efficiency loss, mirroring Figure 8.

**What the formulas suggest.** The analogy to the language-model setting is that $m$ plays the role of model capacity and $d$ corresponds to the unique repeated-data pool $D_{\mathrm{r}}$. In the regimes we simulate in §A.1, the test loss in (7) is non-monotonic in $r$ at fixed $(n, d, m)$, as follows. When $r$ is small, the repeated examples carry little extra weight and the predictor is only weakly affected. When $r$ is too large, the repeated block saturates the rank of $C_u + r^2 C_d$ relative to $C_u + rC_d$, and the test loss returns toward a memorize-and-isolate fixed point. The harmful middle regime appears when the repeated pool is both influential and too large to be harmlessly absorbed. The same formulas suggest that increasing $m$ shifts the peak to larger $d$ in the simulated setting, offering a statistical analogue consistent with the empirical signature in (3). They also suggest weak dependence on the unique-sample count $n$ in this toy setting, consistent with the approximate OT-independence observed in §3.2.

### A.1. Simulations of the linear model

We validate the closed-form theory with direct simulations. Inputs are sampled as $x \sim \mathcal{N}(0, I_p)$ and labels are generated by $y = x^\top \beta$ with $\|\beta\|_2 = 1$. The learner fits OLS on the first $m$ coordinates of $n$ unique samples plus $d$ samples each repeated $r$ times. For each $(m, d, r)$ we evaluate both (7) and the population test loss of the corresponding OLS solve. We report excess test loss over a no-repetition baseline at the same $m$. Figure 5 shows that the closed-form and simulation agree to within numerical precision, that excess loss is non-monotonic in the repeated-pool size $d$ at fixed $m$ and $r$, and that the peak shifts to larger $d$ as $m$ grows. This trend is consistent with the empirical $D_{\mathrm{r}}^{\mathrm{peak}}(N)$ relationship in (3).

We also compute a sample-efficiency analogue of CEG. For each repeated-data run, we estimate the no-repetition unique sample budget $N^{\star}_{\mathrm{clean}}$ required to match its test loss, and define $\mathrm{SE} = N^{\star}_{\mathrm{clean}}/N_{\mathrm{actual}}$, with the repeated block again accounting for $10\%$ of the training budget. Figure 3

shows that SE falls sharply at intermediate $r$ and partially recovers at extreme $r$, mirroring the CEG curve in Figure 8 for language models. Figure 4 shows where the worst repeated-pool size occurs for each model size and repeat count. The closed-form linear model thus captures the same qualitative phenomenon, suggesting that the peak is a generic statistical feature of repeated samples in misspecified models.

## Appendix B. Summarized Related Work

We position our work against three closely connected lines of research. An extended discussion appears in Appendix C.

**Compute-optimal scaling and the Chinchilla scaling law.** Kaplan et al. [26] established the power-law form for transformer loss as a function of compute. Hoffmann et al. [23] corrected the optimal $(N, T)$ allocation, where $N$ is the parameter count and $T$ is the token count. Besiroglu et al. [9], Porian et al. [45] examine the robustness of these fits, Sardana et al. [46] extend them to inference-aware budgets, and Gadre et al. [17] show reliable extrapolation through aggressive over-training. We use this functional form, fitted on no-repetition baselines, as the reference curve for CEG and CEL.

**Deduplication, memorization, and statistical accounts of overfitting.** A large literature studies deduplication and memorization for privacy and contamination reasons [1, 14, 16, 25, 30, 47, 53]. Our setting is distinct in two ways. First, we exclude the evaluation split from training and from all repeated pools by the train/test split, so any harm we observe must come from a distorted effective training distribution. Second, we hold the fraction of repeated tokens fixed and vary only their concentration, isolating the effect of repetition structure. Our theoretical analysis in Appendix A connects to classical results on double descent and benign overfitting [6, 7, 20, 41], specialized to literal sample duplication. We are not aware of prior work using this block-covariance view to analyze literal duplication.

**Repeated data in language model pretraining.** Hernandez et al. [22] repeat a small fraction of training data and observed a non-monotonic test-loss curve. They framed the damage as a reduction in effective parameter count and connected it to degradation of induction heads. Muennighoff et al. [40] study the complementary regime in which the entire corpus is uniformly repeated, and find that up to roughly four epochs are nearly free. Komatsuzaki [29], Maini et al. [36], Xue et al. [58] examine related repetition, rephrasing, and epoch strategies. Recent work further argues that semantic duplication is itself scale-dependent [27] and proposes explicit overfitting penalties for data-constrained scaling [34]. We sit between the Hernandez et al. [22] and Muennighoff et al. [40] regimes. Previous work [22] observed that repeated data can produce non-monotonic held-out loss degradation, and reported the damage as a reduction in effective parameter count. We extend that setting by measuring damage as CEG and CEL: for each repeated-data run, we ask how much compute a no-repetition run on a fitted scaling law would need to reach the same loss. We also run the sweep under Chinchilla-style token and compute budgets while holding the repeated-token fraction fixed at $f = 0.1$ and varying its concentration through $R$, and we fit how the most damaging repeat count shifts with model size.

## Appendix C. Comprehensive related work

We expand here the discussion sketched in §B, organized along five threads.

**Repeated data in language model pretraining.** Our closest predecessor is Hernandez et al. [22], who train transformers with a small fraction of repeated data and observe a non-monotonic test-loss curve. They frame the damage as a reduction in effective parameter count and connect it to mechanistic changes in induction heads [4, 10]. The complementary regime, in which the entire corpus is uniformly repeated, is studied by Muennighoff et al. [40]. They find that up to roughly four epochs are nearly as useful as fresh data and that an additive overfitting term in the Chinchilla loss captures the rest. Xue et al. [58] reach a similar four-epoch threshold from a different angle. Other work in this family includes Komatsuzaki [29], who studies one-pass-vs-multi-pass tradeoffs at smaller scale, and Maini et al. [36], who replaces literal repeats with paraphrases generated by a teacher model. Tirumala et al. [52] and Lesci et al. [31] study how memorization grows with the number of times an example is seen during training. Kazdan et al. [27] argue that semantic duplication is itself scale-dependent, in the sense that larger models recognize more documents as duplicates. Lovelace et al. [34] fit a one-parameter overfitting penalty within Chinchilla scaling that is closely related in spirit to our $\mathrm{CEG}/\mathrm{CEL}$ metric. Our setting differs from all of these in three ways. First, we hold a fixed minority fraction $f = 0.1$ of training tokens repeated, mimicking the residue of imperfect deduplication and complementing the all-or-nothing regimes of Muennighoff et al. [40]. Second, we measure damage in compute-equivalent units derived from a fitted Chinchilla scaling law, sharpening the effective-parameter view of Hernandez et al. [22] into a quantity practitioners directly allocate. Third, we extract a closed-form scaling law for the worst-case configuration as a function of model size, which earlier work has left open.

**Compute-optimal scaling.** The power-law form for transformer loss was established by Kaplan et al. [26] and refined by Hoffmann et al. [23], whose Chinchilla fit is the basis for our no-repetition reference curve. Subsequent work has tested the robustness and generalizability of the Chinchilla functional form. Besiroglu et al. [9] reanalyze the original Chinchilla fits, Porian et al. [45] reconcile competing exponents across studies, Sardana et al. [46] extend the framework to inference-aware budgets, and Gadre et al. [17] show that Chinchilla-style fits extrapolate reliably under aggressive over-training in the regime we use. Bahri et al. [5], Bordelon et al. [11], Caballero et al. [12], Henighan et al. [21] provide alternative parametric forms and theoretical accounts. Tay et al. [51] document the model-family dependence of fitted exponents, supporting our caveat in the Limitations that exponents may shift across architecture families.

**Deduplication, memorization, and benchmark contamination.** Deduplication has been a central tool of pretraining-corpus construction since at least Lee et al. [30], who show that exact and near-duplicate removal improves training efficiency and reduces verbatim regurgitation. Kandpal et al. [25] document a superlinear amplification of privacy risk by duplicates, and Carlini et al. [13, 14] relate memorization to model scale and duplicate count. Semantic deduplication was advanced by Abbas et al. [1] and Tirumala et al. [53], who use embedding clusters to remove near-duplicate documents that elude exact-hashing pipelines. Aggregate surveys include Deng et al. [16] and the data-selection survey of Albalak et al. [3]. A related but distinct concern is benchmark contamination, where evaluation data leaks into training [16, 24, 35, 42, 47]. Schaeffer et al. [47] show that even a single test-set replica can drive measured loss below the no-contamination irreducible floor. We deliberately exclude evaluation documents from training and repeated pools by the fixed train/test split, so the harm we observe comes from a distorted effective training distribution.

**Pretraining corpora and data selection.** Our experiments use FineWeb-Edu-Dedup [44]. Closely related curated web corpora include DataComp-LM [32], Dolma [48], RedPajama-v2 [56], and RefinedWeb [43]. Longpre et al. [33] survey the broader pretrainer's-data landscape. On the data-selection side, Sorscher et al. [49] show that careful pruning can break power-law scaling, Marion et al. [37] study perplexity-based pruning, Xie et al. [57] optimize domain mixtures, and Goyal et al. [18] extend scaling laws to data-quality interventions. Our results complement this view by studying a *negative* form of data selection, asking which residual repetition structures to avoid. The two perspectives are quantitatively connected through the same Chinchilla scaling law, since both ultimately translate dataset choices into a position on a no-intervention scaling curve.

**Statistical accounts of overfitting.** The closed-form analysis in §A sits in the literature on benign overfitting and double descent [2, 6–8, 20, 38, 41, 54], which studies non-monotonic generalization in over-parameterized linear and kernel models. Most of this literature varies model dimension, sample size, or interpolation. Literal verbatim duplication of a subset of observations and the resulting block-diagonal noise covariance (5) are, to our knowledge, novel in this setting. Our derivation isolates the harm from duplication itself, controlling for the change in dataset size that would otherwise confound the effect.

## Appendix D. Setup

We train Qwen3-style models [60] with $N \in \{34, 48, 63, 93, 153, 344\}$M parameters on FineWeb-Edu-Dedup [44]. We sweep the overtraining multiplier $\mathrm{OT} \in \{0.25, 0.5, 1, 2, 4\}$ for all but the larger models due to compute constraints, and sweep the per-document repeat count $R$ on an approximately logarithmic grid from no repeats to as high as $R = 20000$ (we plot up to $R \approx 3000$).

We define this architecture in more detail in Appendix E. Each run is parameterized by a model size $N$ and an overtraining multiplier $\mathrm{OT} \in \{0.25, 0.5, 1, 2, 4\}$ with larger models run on a subset of this grid due to compute constraints. The case $\mathrm{OT} = 1$ corresponds to 20 tokens per parameter, following Chinchilla-style scaling [23], while $\mathrm{OT} > 1$ studies overtraining around that reference point. For a given $(N, \mathrm{OT})$ pair, the total number of training tokens is $T = 20 \cdot \mathrm{OT} \cdot N$. Using the standard dense-transformer estimate of $6N$ FLOPs per token [17, 23, 26, 45, 46], the total training compute is $C = 6NT = 120 \cdot \mathrm{OT} \cdot N^2$. Thus, within each $(N, \mathrm{OT})$ sweep, both the token budget $T$ and compute budget $C$ are fixed.

## Appendix E. Architecture.

We instantiate all models from scratch using the Qwen3 decoder architecture family [60]. Across model sizes, we vary depth, hidden width, and feed-forward width, while holding the remaining architectural settings fixed. All models use rotary position embeddings [50], RMSNorm, SwiGLU feed-forward layers, grouped-query attention, untied input/output embeddings, BF16 training, and FlashAttention-2. The training sequence length is 2048 tokens. The maximum position length is 32768, the vocabulary size is 151670, the attention head dimension is 128, and all models use 32 attention heads and 32 key-value heads. Table 1 reports the exact configurations and parameter counts. Non- embedding parameters exclude both the token embedding matrix and the untied output LM head; total parameters include both.

Table 1: Qwen3-style model configurations used in the experiments.

| Model | Layers | $d_{\text{model}}$ | $d_{\text{ff}}$ | Heads | KV heads | $d_{\text{head}}$ | Train ctx. | Vocab | Non-emb. params | Total params |
|---|---|---|---|---|---|---|---|---|---|---|
| 34M | 3 | 96 | 256 | 32 | 32 | 128 | 2048 | 151670 | 4,941,216 | 34,061,856 |
| 48M | 4 | 128 | 512 | 32 | 32 | 128 | 2048 | 151670 | 9,177,216 | 48,004,736 |
| 63M | 5 | 160 | 512 | 32 | 32 | 128 | 2048 | 151670 | 14,339,040 | 62,873,440 |
| 93M | 6 | 224 | 768 | 32 | 32 | 128 | 2048 | 151670 | 25,121,120 | 93,069,280 |
| 153M | 9 | 320 | 1024 | 32 | 32 | 128 | 2048 | 151670 | 56,041,664 | 153,110,464 |
| 344M | 14 | 576 | 1536 | 32 | 32 | 128 | 2048 | 151670 | 169,299,776 | 344,023,616 |

## Appendix F. Repeated-pool construction details

For each run, we first split FineWeb-Edu-Dedup into training and held-out evaluation documents using train/test split seed 0. The evaluation split is constructed before any repeated pools are selected, so evaluation documents are excluded from both the repeated and non-repeated training streams.

Documents are tokenized with the Qwen3 [60] tokenizer, truncated to the training sequence length, and assigned EOS tokens before token counts are computed. Let $T$ be the target number of training tokens for a run, $f = 0.1$ the target repeated-token fraction, and $R$ the number of times each repeated document is replayed. The target unique repeated-pool size is

$$D_{\text{r}}^{\star} = \frac{fT}{R},$$

and the target non-repeated budget is $(1 - f)T$.

Documents are selected without replacement from the training split. We form a seeded random ordering of training documents using the run's shuffle seed. The repeated pool is the first prefix of this ordering whose cumulative token count reaches $D_{\text{r}}^{\star}$. The non-repeated pool is then selected from the immediately following documents until its cumulative token count reaches $(1-f)T$. Thus selection is uniform over documents through a seeded shuffle, not token-weighted sampling, and the repeated documents are disjoint from the non- repeated documents.

Because cutoffs occur at document boundaries, the realized unique repeated-pool size $\widehat{D}_{\text{r}}$ and realized repeated-token fraction $\widehat{f}$ are approximate. If $L_{\max}$ is the maximum tokenized document length after truncation and EOS insertion, then

$$D_{\text{r}}^{\star} \leq \widehat{D}_{\text{r}} < D_{\text{r}}^{\star} + L_{\max}.$$

The non-repeated pool has the same one-document overshoot bound. In our preprocessing, $L_{\max} \leq 2049$ because documents are truncated to length 2048 and an EOS token may be appended after truncation. Each document in the repeated pool is then inserted exactly $R$ times, each non-repeated document is inserted once, and the resulting document-index list is shuffled before training.

Throughout the paper, $D_{\text{r}} = fT/R$ denotes the target unique repeated-pool size. The approximation $fT \approx RD_{\text{r}}$ reflects this document-boundary rounding.

## Appendix G. Linear regression with repeated samples

We derive the formulas used in Section A. Let the original training set contain $n$ unique examples and $d$ repeatable examples. The repeatable block is duplicated $r$ times, so the expanded training set has

$$N = n + rd$$

rows. The learner observes only the first $m$ coordinates of each input. Write the expanded observed-feature matrix as $X_{\text{in}}$ and the unobserved-feature matrix as $X_{\text{out}}$. Labels are noiseless:

$$y = X_{\text{in}}\beta_{\text{in}} + X_{\text{out}}\beta_{\text{out}}.$$

The restricted OLS estimator is

$$\hat{\beta}_{\text{in}} = (X_{\text{in}}^\top X_{\text{in}})^{-1} X_{\text{in}}^\top y = \beta_{\text{in}} + a_r,$$

where

$$a_r = (X_{\text{in}}^\top X_{\text{in}})^{-1} X_{\text{in}}^\top X_{\text{out}}\beta_{\text{out}}.$$

Thus $a_r$ is the aliasing term caused by fitting the unobserved part of the signal using the observed coordinates.

Let

$$C_u = X_{u,\text{in}}^\top X_{u,\text{in}}, \qquad C_d = X_{d,\text{in}}^\top X_{d,\text{in}}.$$

Then

$$X_{\text{in}}^\top X_{\text{in}} = C_u + rC_d.$$

Now condition on $X_{\text{in}}$ and take expectation over the unobserved features. Let

$$z = X_{\text{out}}\beta_{\text{out}}.$$

Because repeated rows share the same unobserved coordinates, the covariance of $z$ over the expanded training set is not proportional to the identity. Instead,

$$\mathbb{E}[zz^\top \mid X_{\text{in}}] = \|\beta_{\text{out}}\|_2^2\, \Sigma_r, \qquad \Sigma_r \;=\; I_n \;\oplus\; \bigoplus_{i=1}^{d} \mathbf{1}_r \mathbf{1}_r^\top,$$

where $\mathbf{1}_r \in \mathbb{R}^r$ is the all-ones vector and the $i$-th repeated block is the rank-one $r \times r$ matrix $\mathbf{1}_r\mathbf{1}_r^\top$. The unique block contributes the $n \times n$ identity (independent unobserved features); each repeated block has every entry equal to every other (the $r$ copies of document $i$ share a single $x_{\text{out},i}$). The full matrix is $(n+rd) \times (n+rd)$ and block-diagonal; in particular, distinct repeated documents are uncorrelated. Therefore,

$$X_{\text{in}}^\top \Sigma_r X_{\text{in}} = C_u + r^2 C_d.$$

For training loss, let

$$H = X_{\text{in}}(X_{\text{in}}^\top X_{\text{in}})^{-1} X_{\text{in}}^\top$$

be the projection matrix onto the observed-feature span. The residual is $(I - H)z$, so

$$\mathbb{E}[L_{\text{train}} \mid X_{\text{in}}] = \frac{1}{N}\mathbb{E}[z^\top (I-H) z \mid X_{\text{in}}].$$

Using the covariance above,

$$\mathbb{E}[L_{\text{train}} \mid X_{\text{in}}] = \frac{\|\beta_{\text{out}}\|_2^2}{N}\, \operatorname{tr}((I-H)\Sigma_r).$$

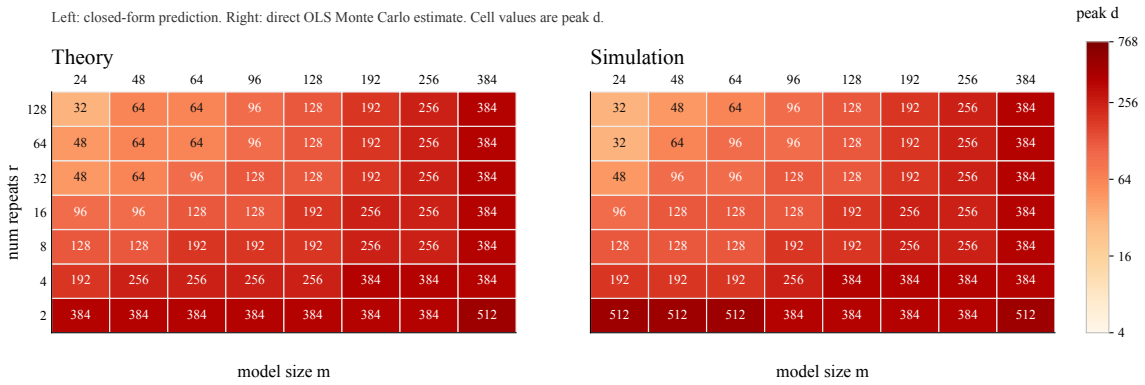


Figure 4: Repeated-pool size at peak excess test loss.

Since $\mathrm{tr}(\Sigma_r) = N$ and

$$\mathrm{tr}(H\Sigma_r) = \mathrm{tr}\left((C_u + r^2 C_d)(C_u + rC_d)^{-1}\right),$$

we obtain

$$\mathbb{E}[L_{\text{train}} \mid X_{\text{in}}] = \frac{\|\beta_{\text{out}}\|_2^2}{N}\left[N - \mathrm{tr}\left((C_u + r^2 C_d)(C_u + rC_d)^{-1}\right)\right].$$

For test loss, take a fresh example $(x_{\text{in}}, x_{\text{out}}) \sim \mathcal{N}(0, I_p)$ from the population, with $x_{\text{in}}$ and $x_{\text{out}}$ independent under the isotropic assumption. The prediction error is

$$x_{\text{out}}^\top \beta_{\text{out}} - x_{\text{in}}^\top a_r.$$

The two terms are independent and mean zero (the cross term vanishes by independence and zero mean of $x_{\text{in}}$), so

$$\mathbb{E}[L_{\text{test}} \mid X_{\text{in}}] = \|\beta_{\text{out}}\|_2^2 + \mathbb{E}[\|a_r\|_2^2 \mid X_{\text{in}}].$$

Substituting the expression for $a_r$ gives

$$\mathbb{E}[\|a_r\|_2^2 \mid X_{\text{in}}] = \|\beta_{\text{out}}\|_2^2 \,\mathrm{tr}\left((C_u + rC_d)^{-1}(C_u + r^2 C_d)(C_u + rC_d)^{-1}\right),$$

and therefore

$$\mathbb{E}[L_{\text{test}} \mid X_{\text{in}}] = \|\beta_{\text{out}}\|_2^2\left[1 + \mathrm{tr}\left((C_u + rC_d)^{-1}(C_u + r^2 C_d)(C_u + rC_d)^{-1}\right)\right].$$

## Appendix H. Training and evaluation details

**Architecture.** We use Qwen3-style decoder-only transformers [59–61] with rotary position embeddings [50], RMSNorm, SwiGLU feed-forward layers, grouped-query attention, and the Qwen3 tokenizer. Six parameter counts are used, $N \in \{34, 48, 63, 93, 153, 344\}$M, obtained by scaling depth and width approximately uniformly. We compute $C = 6NT$ FLOPs per the standard dense-transformer estimate [23, 26].

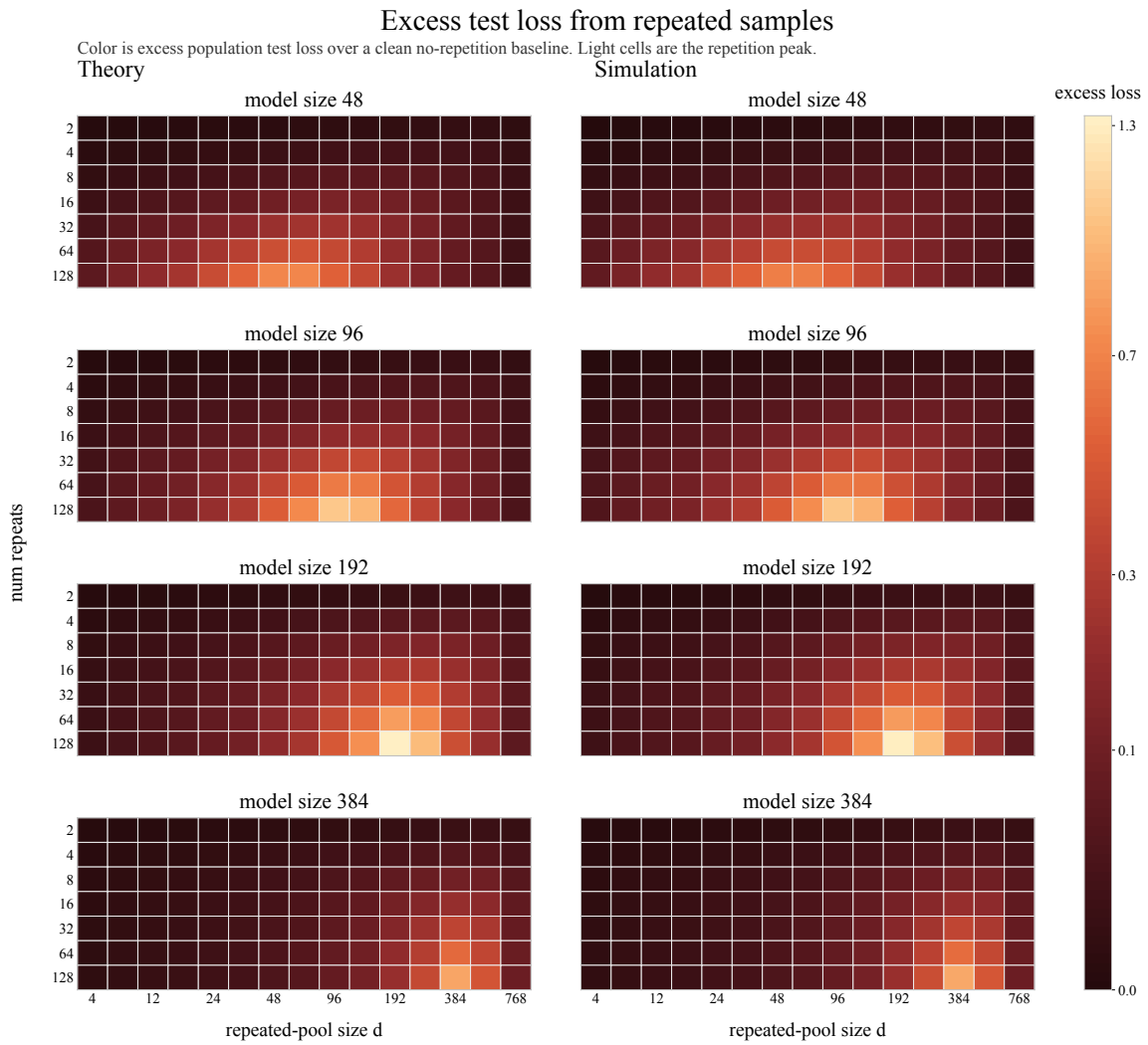


Figure 5: To isolate the peak location, we fix $(m, r)$ and report the repeated-pool size $d$ that maximizes excess test loss. The same trend appears in both the closed-form risk and direct OLS simulations: larger-capacity models peak at larger repeated pools, while higher repeat counts shift the peak toward smaller pools. We note that theory and simulation agree up to the resolution of the $d$ grid.

**Optimizer and schedule.** All runs use AdamW [28] with the fused PyTorch implementation (`adamw_torch_fused`), $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay $0.01$, and gradient clipping at $1.0$. The learning rate follows a cosine schedule with warmup ratio $0.2$. The peak learning rate is derived from a base learning rate of $10^{-6}$ and the computed optimizer-step token count, rather than tuned separately per model size. Sequence length is 2048 throughout. Training uses BF16 weights, FlashAttention-2, and torch compilation.

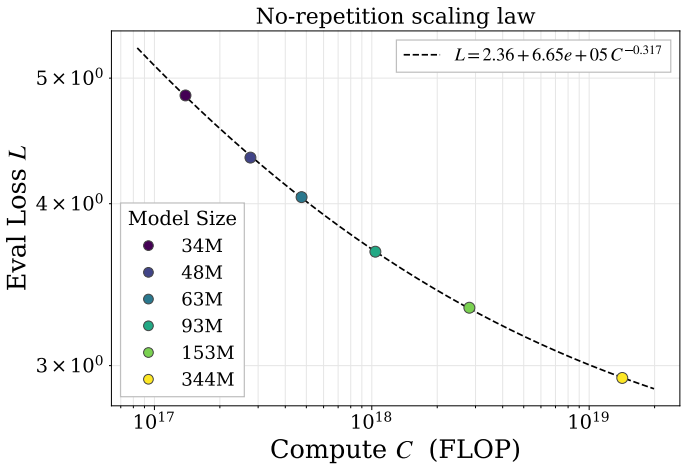


Figure 6: No-repetition scaling law fit. We fit $L(C) = E + KC^{-\gamma}$ to the six OT=1, $R = 1$ baselines. This scaling law converts any eval loss into the equivalent no-repetition compute, enabling the CEG and CEL metrics.

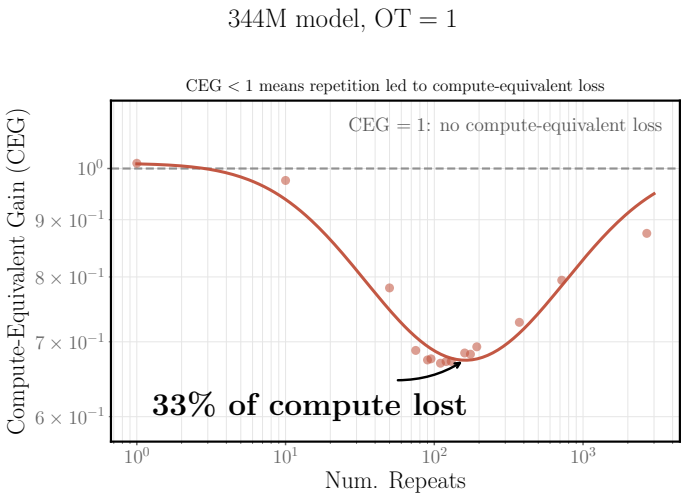


Figure 7: At $R$ near 100, CEL rises to roughly 0.33, meaning the run reaches the loss of a run without repetitions trained with only two-thirds of the FLOPs [44, 60].

**Data and evaluation.** The source corpus is FineWeb-Edu-Dedup from the HuggingFaceTB SmolLM corpus Penedo et al. [44]. We split the corpus once with train/test split seed 0, holding out approximately 150M tokens for evaluation before constructing any repeated pools. Thus evaluation documents are excluded from both the non-repeated training stream and the repeated pool. Documents are tokenized with the Qwen3 tokenizer [60], truncated to length 2048, and assigned EOS tokens. For each repeated-data run, the repeated pool is selected at document granularity to contribute approximately $fT/R$ unique tokens. These documents are then replayed $R$ times, combined with

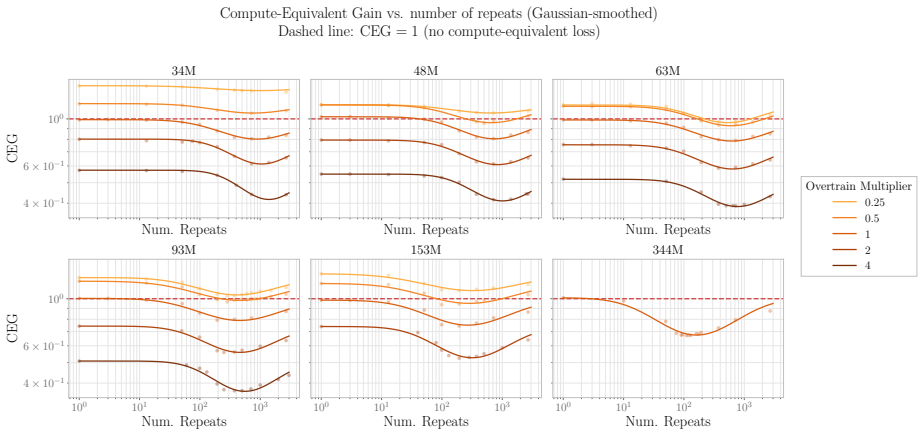


Figure 8: Compute-Equivalent Gain as a function of repeat count, by model size and overtraining multiplier. CEG $= 1$ (dashed line) matches the no-repetition reference. CEG falls at intermediate repeat counts and partially recovers at large ones, revealing a worst-case repetition structure for each sweep. Damage grows with model size. The Qwen3-style 344M-parameter model reaches CEG $\approx 0.67$, corresponding to CEL $\approx 0.33$.

non-repeated documents contributing approximately $(1 - f)T$ tokens, and shuffled into the final training stream. This makes the repeated-token fraction approximately $f = 0.1$ while varying the concentration of those repeated tokens.

**Sweep grid.** We train no-repetition baselines and repeated-data sweeps for each completed $(N, \mathrm{OT})$ cell. The completed analysis grid contains 25 cells: all five overtraining multipliers for 34M, 48M, 63M, and 93M; four multipliers through $\mathrm{OT} = 2$ for 153M; and $\mathrm{OT} = 1$ for 344M. Repeat counts are swept on an irregular, approximately logarithmic grid spanning from $R = 1$ to as high as $R = 20000$, subject to the repeated pool containing at least one document. Each completed cell is a single training run.

**Peak fitting.** For each completed $(N, \mathrm{OT})$ sweep, we first compute the fractional eval-loss increase relative to the corresponding no-repetition baseline, $\eta(R) = (L(R) - L_{\mathrm{base}})/L_{\mathrm{base}}$. We then fit a three-parameter Gaussian in $\log_{10} R$ to the non-baseline points ($R > 1$):

$$\eta(R) = A \exp\left( -\frac{(\log_{10} R - \log_{10} \mu)^2}{2\sigma^2} \right).$$

The fitted peak repeat count is $R^{\mathrm{peak}} = \mu$, and the corresponding repeated-pool size is computed from $D_{\mathrm{r}}^{\mathrm{peak}} = 2\mathrm{OT}N/R^{\mathrm{peak}}$. Power-law fits in (3)–(4) are ordinary least-squares regressions in log-log space over the completed sweeps.

**Scaling-law fitting.** The no-repetition scaling law $L(C) = E + KC^{-\gamma}$ in §3.3 is fit on the six $\mathrm{OT} = 1$, $R = 1$ baselines. We fit in log-loss space using nonlinear least squares: first estimating an initial $K$ and $\gamma$ from a log-log linear fit without a loss floor, then refitting $E$, $K$, and $\gamma$ jointly with $E$ free. This gives (3.3). The fitted curve matches the six no-repetition baselines to within about $0.015$ nats, and we discuss sensitivity to this three-parameter fit in §3.3.

The Chinchilla budget identity $C = 6NT = 120 \cdot \mathrm{OT} \cdot N^2$ ties model size, total tokens $T$, and total compute $C$ together [17, 26, 45, 46]. This lets us vary repetition structure inside an otherwise fixed training budget.

**Reported evaluation loss.** All reported eval losses are final-checkpoint losses. After each run reaches its target token budget $T = 20 \cdot \mathrm{OT} \cdot N$, up to document-boundary rounding from the sampling procedure, we evaluate the final checkpoint once on the fixed held-out split and use that value in all figures, peak fits, and scaling-law fits. We use the same rule for repeated-data runs and no-repetition baselines. We do not report the best validation checkpoint or an average over checkpoints; intermediate evaluations are used only for monitoring.

## Appendix I. Interpreting $\mathrm{CEG} > 1$

The CEG ratio in (3.3) is defined relative to our fitted no-repetition scaling law. It is not an oracle optimum over all possible model sizes, token budgets, optimizers, and data mixtures. Therefore $\mathrm{CEG} > 1$ can occur when a run achieves lower loss than the fitted no-repetition reference curve predicts at its actual compute. We interpret such values as being above this fitted reference, not as a universal claim that the run is optimally compute saving.

One reason this can happen is that $\mathrm{OT} = 1$ uses the Chinchilla-style rule of 20 tokens per parameter. That rule was estimated in a different setting, with a different model family, optimizer stack, tokenizer, and data mixture. The optimal token-per-parameter ratio for our Qwen3-style [60] models on FineWeb-Edu-Dedup [44] may therefore differ from 20. We do not attempt to find the globally optimal $(N, T)$ allocation for this model family, because the goal of the paper is to compare repetition structures at fixed budget and to convert their losses through a consistent no-repetition reference.

For the repeated-data comparisons, what matters is that all runs are evaluated against the same fitted no-repetition reference, and that each repeated-data run is also compared to the no-repetition baseline at the same $(N, \mathrm{OT})$ budget. Values below one imply positive CEL relative to this reference. Values above one imply negative CEL and should be read as a calibration effect of the fitted reference curve.

## Appendix J. Sensitivity to the loss floor

Equation (3.3) contains the factor $(L - E)^{-1/\gamma}$, which diverges as $L$ approaches the fitted floor $E$. With $\gamma \approx 0.32$, a $1\%$ shift in $(L - E)$ produces a $\approx 3\%$ relative shift in CEG. The 344M peak run sits at $L - E \approx 0.65$ nats, so the headline 33% number is robust to leave-one-out perturbations of the scaling-law fit. Smaller models live closer to $E$ and inherit larger uncertainty. The scaling law is fit on six points, at the boundary of identifiability for the three-parameter Chinchilla form [9, 45]. We therefore treat the absolute CEG and CEL values as point estimates and the qualitative non-monotonicity as the robust finding.

## Appendix K. Limitations

Our experiments are small relative to frontier pretraining: the largest model has 344M parameters, and all runs use one Qwen3-style architecture family, one tokenizer, one corpus, and a fixed

repeated-token fraction $f = 0.1$. Each $(N, \mathrm{OT}, R)$ cell is a single training run, so we do not estimate seed-level variance; the evidence for robustness comes from consistency across sweeps, not repeated random restarts. The larger-model grid is incomplete because of compute constraints, with 153M run through $\mathrm{OT} = 2$ and 344M run only at $\mathrm{OT} = 1$. The repeat-count sweep extends to $R = 20000$; the log-Gaussian peak fits use the full range, though plots display only up to $R \approx 3000$ for readability. Finally, CEG and CEL depend on a three-parameter no-repetition scaling law fit to six $\mathrm{OT} = 1$ baselines, so their absolute values should be treated as point estimates. The linear-regression model in Appendix A explains one mechanism by which repetition can hurt, but it leaves out many features of real language models, including attention, depth, optimization dynamics, and discrete tokens. It should be read as an explanatory toy model, not a quantitative model of transformer pretraining.

## Appendix L. Broader Impacts

This work studies how repeated training data can reduce the compute efficiency of language-model pretraining. Better measurement of repetition damage may help practitioners spend less compute and energy on runs whose data mixture is poorly structured. The same analysis could also be used to justify more aggressive data filtering, so care is needed to avoid removing useful minority-domain or low-resource-language data solely because it appears repetitive.

## Appendix M. Experiments Compute Resource

All experiments were run on GPU clusters using BF16 training and FlashAttention-2. The main experimental grid consists of Qwen3-style models from 34M to 344M parameters, trained across repeated-data sweeps and no-repetition baselines as described in Appendix H. The largest individual runs are the 344M-parameter models at $\mathrm{OT} = 1$, with total training compute estimated by $C = 6NT$.