INTERNAL DATA REPETITION DESTROYS LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

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 era, using a fitted no-repetition scaling law to report Compute-Equivalent Gain and Compute-Equivalent Loss. We show that under this modernized paradigm, repetition damage is systematic in three ways. First, when holding the amount of 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 peak location is well described by a power law in model size; within the range studied, this trend shows that the most damaging repeated-data pool grows faster than the total training-token budget at a fixed overtraining multiplier. 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 fitted peak for a Qwen3-style 344M-parameter model at $\mathrm{OT} = 1$ matches the loss predicted by a no-repetition calibration 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, quantifying how 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 and repeat structure of duplicates in pretraining corpora.

1 INTRODUCTION

Pretraining has entered a data-constrained regime. The high-quality public text corpora used for frontier training have been exhausted (Villalobos et al., 2024; Longpre et al., 2024), forcing multi-epoch training. To maintain data efficiency in pretraining, flagship corpora such as FineWeb-Edu, DataComp-LM, Dolma, and RedPajama-v2 (Penedo et al., 2024; Li et al., 2024; Soldaini et al., 2024; Weber et al., 2024) have implemented aggressive deduplication and filtering. Yet aggressive deduplication is not perfect deduplication (Lee et al., 2022; Abbas et al., 2023; Tirumala et al., 2023): pretraining streams retain near-duplicate documents, paraphrased templates, and semantically redundant web pages, and as scale grows the meaning of "duplicate" itself shifts (Kazdan et al., 2026). In this paper, we isolate one controlled case: exact document-level replay of a selected repeated pool. We precisely quantify the cost incurred by exact repeats during pretraining.

The closest earlier controlled study, Hernandez et al. (2022), established that repeating a small training-data fraction non-monotonically degrades held-out loss and framed the damage as a reduction in *effective parameter count*, an outdated measurement that tried to quantify performance in terms of parameters rather than compute, where all models were trained on 100B tokens irrespective of parameter count. This left the larger models undertrained while the smaller ones were grossly overtrained. Hernandez et al. (2022)'s work predated the widespread use of compute-optimal scaling (Hoffmann et al., 2022), which provided a clean compute axis for scaling-law comparisons. Modern practitioners measure the cost of training in FLOPs, not parameters. We modernize the study of memorization scaling laws by measuring dam-

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

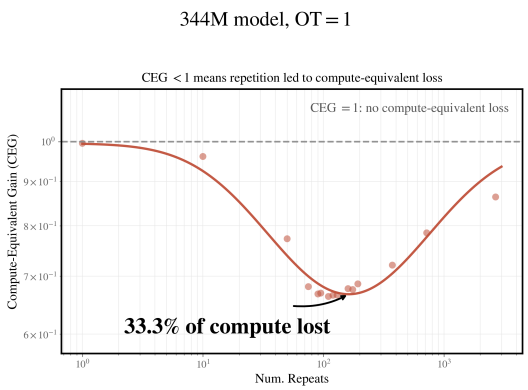

Figure 1: Compute-Equivalent Gain (CEG) as a function of the per-document repeat count R at fixed training compute for the Qwen3-style 344M-parameter model trained on FineWeb-Edu-Dedup at the Chinchilla-style multiplier $\mathrm{OT} = 1$. CEG is the ratio of the no-repetition compute that would reach the achieved loss to the compute actually spent. $\mathrm{CEG} = 1$ is when there is no gain or loss relative to the no repetition baseline. The sampled peak at $R = 110$ has $\mathrm{CEL} \approx 0.337$, while the smoothed peak has $\mathrm{CEL} \approx 0.333$, meaning the run reaches the loss of a run without repetitions trained with only two-thirds of the FLOPs Yang et al. (2025a); Penedo et al. (2024).

age via *Compute-Equivalent Gain (*CEG*)*, following the compute-equivalent gains framework of Gundlach et al. (2025); Davidson et al. (2023), and Meta Superintelligence Labs (2026). For a repeated-data run with loss L and actual compute C_{actual}, CEG is the no-repetition compute required to reach L divided by C_{actual}. We define *Compute-Equivalent Loss* as $\mathrm{CEL} = 1 - \mathrm{CEG}$. Thus, $\mathrm{CEG} = 1$ matches the no-repetition baseline, while $\mathrm{CEG} < 1$ indicates compute-equivalent loss.

To modernize the study of repetition damage, we train Qwen3-style models (Yang et al., 2025a) with $N \in \{34, 48, 63, 93, 153, 344\}$M parameters on FineWeb-Edu-Dedup (Penedo et al., 2024). We sweep the overtraining multiplier $\mathrm{OT} \in \{0.25, 0.5, 1, 2, 4\}$ and the per-document repeat count R on an approximately logarithmic grid from no repeats to as high as $R = 20000$ (we plot up to $R \approx 3000$). In the main sweeps, we fix the repeated-token fraction at $f = 0.1$: 90% of training tokens come from non-repeated documents, and the remaining 10% come from a smaller document pool replayed R times. This fraction is large enough to produce measurable damage while keeping the bulk of training on unique data, and matches the repeated-token fraction used by Hernandez et al. (2022), enabling a direct comparison.

The Chinchilla budget identity $C \approx 6NT = 120 \cdot \mathrm{OT} \cdot N^2$ ties model size, total tokens T, and total compute C together (Gadre et al., 2025; Porian et al., 2024; Kaplan et al., 2020; Sardana et al., 2024). This lets us vary repetition structure inside an otherwise fixed training budget, separating the effect of repeat concentration from total compute. For example, fixing the memorization compute budget at 10%, we can repeat either 1% of the pretraining corpus 10 times, or we can repeat 0.01% of the pretraining corpus 1000 times. We refer to this pairing of repeated-pool size and repeat count as a *repetition structure*. We find that the most damaging repetition structure is a function of the model's parameter count N, and we translate the resulting loss increases into Compute-Equivalent Gain and Compute-Equivalent Loss using a fitted no-repetition scaling law. Under this fitted frontier, the most damaging repeat count at our largest scale has $\mathrm{CEG} \approx 0.67$. A misspecified linear regression with verbatim duplicates reproduces the same qualitative non-monotonicity in closed form, suggesting that such peaks can arise from the statistical structure of duplicated samples rather than from any attention, depth, or optimizer-specific effects particular to language model training.

Contributions. (i) We quantify repeated-data damage in compute-equivalent units using a fitted no-repetition Chinchilla scaling law (§4.3). At 344M parameters and $\mathrm{OT} = 1$ on FineWeb-Edu-Dedup Yang et al. (2025a); Penedo et al. (2024), the most damaging repeat count has $\mathrm{CEL} \approx 0.33$. (ii) We identify the intermediate-R regime where damage peaks (§4.1) and show that its location follows the power law $R^{\mathrm{peak}} \propto N^{-0.872}$ (§4.2), summarizing the worst-case configuration within the measured range. (iii) We derive closed-form train and test losses for a mis-

specified linear regression with verbatim duplicates (§4.4) and recover the same non-monotonic peak in simulation, giving a simple statistical analogue for the empirical pattern observed in the language-model sweeps under exact document replay at fixed repeated-token fraction.

2 Related Work

We position our work against three related lines of research; Appendix A extends the discussion.

Compute-optimal scaling and the Chinchilla scaling law. Kaplan et al. (2020) established the power-law form for transformer loss as a function of compute. Hoffmann et al. (2022) corrected the optimal (N, T) allocation, where N is the parameter count and T is the token count. Porian et al. (2024); Besiroglu et al. (2024) examine the robustness of these fits, Sardana et al. (2024) extend them to inference-aware budgets, and Gadre et al. (2025) show reliable extrapolation through aggressive over-training. We use this functional form as our fitted no-repetition reference curve.

Deduplication, memorization, and statistical accounts of overfitting. A large literature studies deduplication and memorization for privacy and contamination reasons (Lee et al., 2022; Kandpal et al., 2022; Carlini et al., 2023; Abbas et al., 2023; Tirumala et al., 2023; Deng et al., 2024; Schaeffer et al., 2026). 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 connects to classical results on double descent and benign overfitting (Belkin et al., 2019; Hastie et al., 2022; Bartlett et al., 2020; Nakkiran et al., 2020), 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. (2022) 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. (2023) study the complementary regime in which the entire corpus is uniformly repeated, and find that up to roughly four epochs are nearly free. Xue et al. (2023); Maini et al. (2024); Komatsuzaki (2019) examine related repetition, rephrasing, and epoch strategies. Recent work further argues that semantic duplication is itself scale-dependent (Kazdan et al., 2026) and proposes explicit overfitting penalties for data-constrained scaling (Lovelace et al., 2026). We sit between these two regimes, and extend Hernandez et al. (2022) by measuring damage as CEG and CEL rather than as a reduction in effective parameter count: for each repeated-data run, we ask how much compute a no-repetition run 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.

3 Methods

Repeated data is increasingly hard to avoid during pretraining. Thus, it becomes essential to understand which configurations of repeated data are dangerous and how much compute they waste. In the main sweeps, we fix the repeated-token fraction–or equivalently, repeated-token compute fraction–at $f = 0.1$ and vary only its structure. By holding the fraction of tokens dedicated to repeats constant, we isolate the impact of repetition *structure* from that of repetition *amount*, letting us identify the most harmful configurations at fixed compute.

Setup. We train Qwen3-style decoder-only transformers (Yang et al., 2025a; Vaswani et al., 2017; Su et al., 2024) on FineWeb-Edu-Dedup (Penedo et al., 2024), using six parameter counts $N \in \{34, 48, 63, 93, 153, 344\}$M. We define this architecture in more detail in Appendix B. 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. Setting $\mathrm{OT} = 1$ corresponds to 20 tokens per parameter, following Chinchilla-style scaling (Hoffmann et al., 2022), while $\mathrm{OT} > 1$ implies overtraining relative to the compute-optimal token amount. For a given (N, 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 (Kaplan et al., 2020; Hoffmann et al., 2022; Sardana et al., 2024; Porian et al., 2024; Gadre et al., 2025), the total training compute is

$$C = 6NT = 120 \cdot \mathrm{OT} \cdot N^2. \tag{1}$$

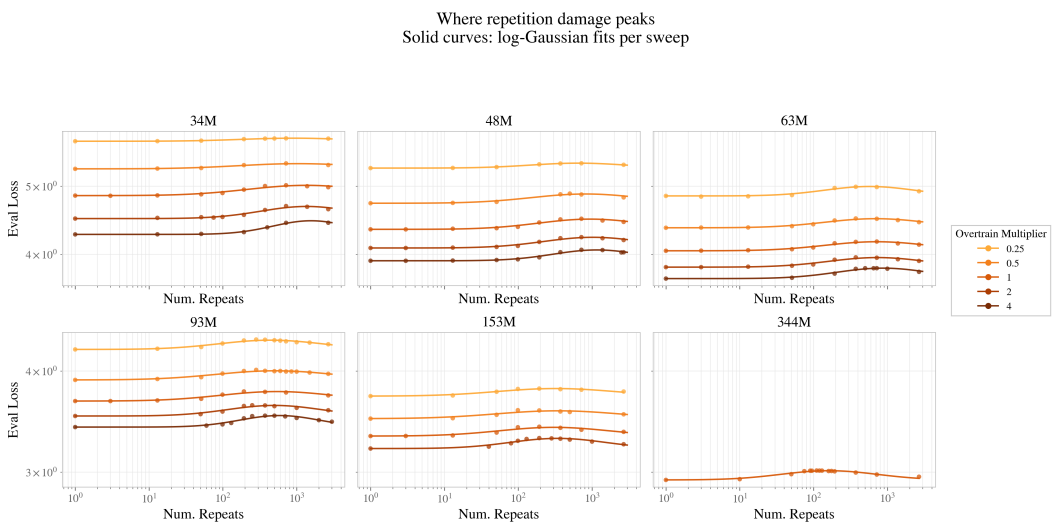

Figure 2: Gaussian fits to eval loss as a function of repeat count. Each panel fixes a model size N, and each curve corresponds to an overtraining multiplier OT. The fitted peak gives R^{peak}, the repeat count at which eval loss is largest for that (N, OT) sweep. Across all model sizes, eval loss is maximized at an intermediate repeat count.

Thus, within each (N, OT) sweep, both the token budget T and compute budget C are fixed.

Repeated-pool construction. For each repeated-data run, $(1-f)T$ tokens are drawn from non-repeated documents. Let D_{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

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

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_{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 pretraining setup—and identify the configurations that cause the most damage. See Appendix C for more details on the sampling protocol.

Evaluation and baselines. We evaluate the main-grid models 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, OT) sweep, we also train a no-repetition baseline. The original per-sweep baselines and additional baseline replications serve two purposes. First, the baseline at the same (N, OT) gives the per-sweep reference for measuring the fractional eval-loss increase caused by repetition. Second, three $\text{OT} = 1$ no-repetition runs at each of six model sizes calibrate the no-repetition Chinchilla scaling law $L(C) = E + KC^{-\gamma}$ used in §4.3 to convert eval loss into CEG and CEL. Additional implementation and evaluation details are given in Appendix F.

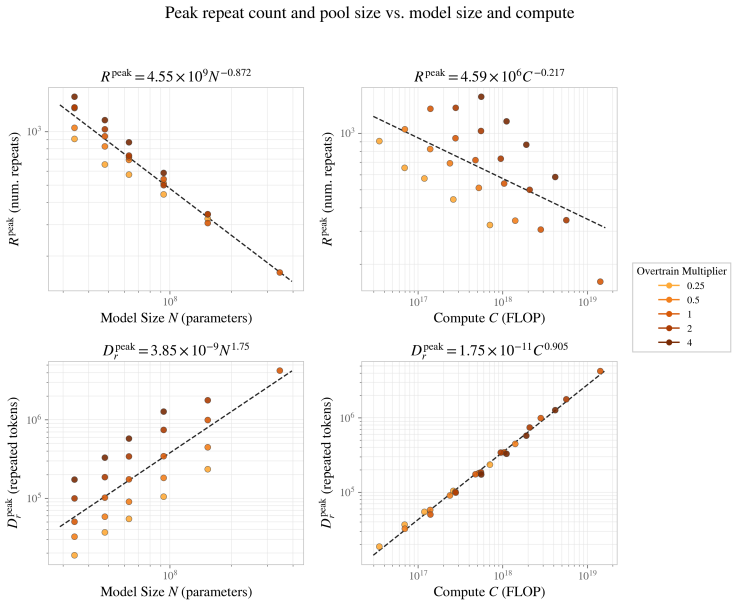

Figure 3: Scaling laws for the peak-damage regime. Top row, the repeat count at peak eval loss R^{peak} decreases with model size (left) and compute (right). Bottom row, the repeated-pool size at peak eval loss $D_{\text{r}}^{\text{peak}}$ increases with both. Lines show separate descriptive fits to 25 cells; intervals condition on the fitted peaks.

4 Results

We characterize how eval loss depends on the structure of the repeated subset (§4.1), extract a model-size scaling law for the worst-case configuration (§4.2), translate the resulting loss differences into CEG and CEL via the no-repetition Chinchilla scaling law (§4.3), and explain the same non-monotonicity in a closed-form linear-regression model (§4.4–§4.5).

4.1 Eval loss is non-monotonic in the repeat count

For each (N, 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 2 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 §4.2 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. (2022) 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 §4.3. 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, though the recovery at large R is flatter in some sweeps. §4.4 offers a statistical mechanism for why such a peak can arise.

4.2 Peak locations are consistent with a power-law trend in model size

§4.1 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

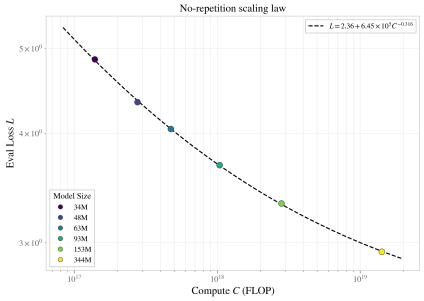

Figure 4: No-repetition scaling law fit. We fit $L(C) = E + KC^{-\gamma}$ to three-run mean losses at six model sizes with $\mathrm{OT} = 1$, $R = 1$; bars show sample standard deviations. This scaling law converts an eval loss above the fitted floor into the equivalent no-repetition compute, enabling the CEG and CEL metrics.

size. For each (N, OT) pair, we fit the fractional excess loss $\eta = (L - L_{\mathrm{base}})/L_{\mathrm{base}}$ at $R > 1$, with $x = \log_{10} R$,

$$\eta(x) = A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad R^{\mathrm{peak}} = 10^{\mu}. \tag{3}$$

This fit captures the single peak we observe in log-repeat space. We then fit power laws to the estimated R^{peak} values across the completed (N, OT) grid and convert them to repeated-pool sizes using equation 2, giving:

$$R^{\mathrm{peak}} = 4.55 \times 10^{9} N^{-0.872}, \qquad D_{\mathrm{r}}^{\mathrm{peak}} = 3.85 \times 10^{-9} N^{1.75}, \tag{4}$$

$$R^{\mathrm{peak}} = 4.59 \times 10^{6} C^{-0.217}, \qquad D_{\mathrm{r}}^{\mathrm{peak}} = 1.75 \times 10^{-11} C^{0.905}. \tag{5}$$

Figure 3 shows four separate log–log fits across 25 (N, OT) cells, using exact total parameter counts. The model-size exponent for R^{peak} is -0.872, with a conditional cell-bootstrap interval of $[-0.962, -0.773]$. These fits summarize the measured range; they do not establish extrapolation to larger models or a scaling law for memorization capacity. Appendix I gives the fitting and resampling details.

Observed trend. Within the measured range, larger models tend to reach their largest loss increase at fewer repeats of a larger repeated pool. At $\mathrm{OT} = 1$, the fitted 34M peak is at $R \approx 1353$, with $D_{\mathrm{r}} \approx 5.04 \times 10^4$ tokens; the 344M peak is at $R \approx 162$, with $D_{\mathrm{r}} \approx 4.25 \times 10^6$ tokens. The available OT sweeps cluster around a similar trend in N, but limited seed replication leaves peak-location uncertainty. Additional $f = 0.05$ sweeps and a 93M seed/shuffle/split replication appear in Appendix I.

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

The 2 to 4% loss increases reported in §4.1 and §4.2 translate into different compute gaps depending on where the run sits on the no-repetition Chinchilla curve. Hernandez et al. (2022) 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 mean losses of three $\mathrm{OT}{=}1$, $R{=}1$ runs at each of six model sizes (Hoffmann et al., 2022; Gadre et al., 2025; Porian et al., 2024), obtaining

$$L(C) = 2.36 + 6.45 \times 10^{5} C^{-0.316}. \tag{6}$$

The fit uses nominal $T = 20N$ budgets and is shown in Figure 4. Here E is the fitted asymptotic loss floor, 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

$$\mathrm{CEG} = \frac{C^\star(L)}{C_{\mathrm{actual}}}, \qquad \mathrm{CEL} = 1 - \mathrm{CEG}. \tag{7}$$

$\mathrm{CEG} = 1$ 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.

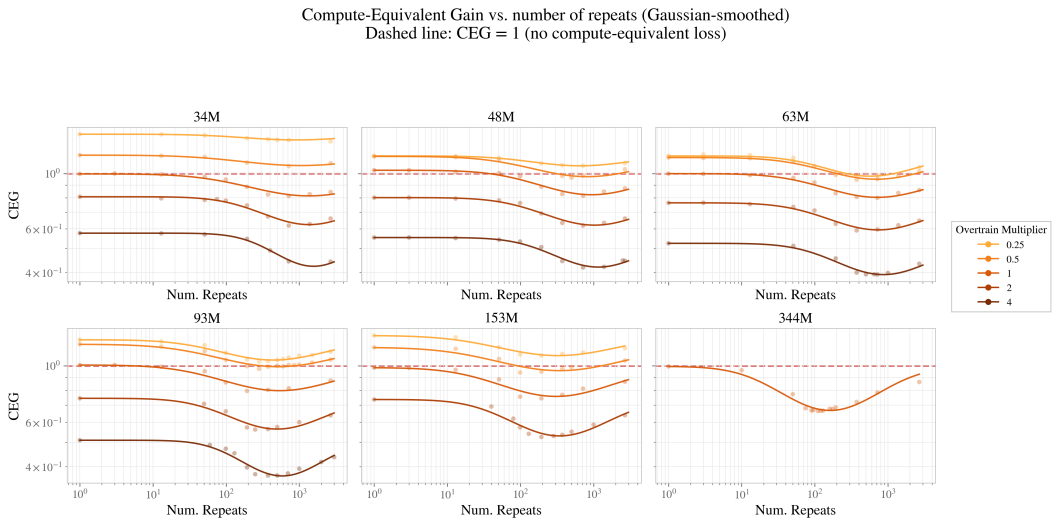

Figure 5: Compute-Equivalent Gain as a function of repeat count, by model size and overtraining multiplier. $\mathrm{CEG} = 1$ (dashed line) matches the no-repetition reference. CEG falls at intermediate repeat counts and partially recovers at large ones, revealing a worst-case repetition structure for each sweep. At $\mathrm{OT} = 1$, the peak compute-equivalent loss grows across the measured sizes. The Qwen3-style 344M-parameter model reaches $\mathrm{CEG} \approx 0.67$, corresponding to $\mathrm{CEL} \approx 0.33$.

Figure 5 shows CEG as a function of R. At OT=1, the worst repeat settings increase CEL to $0.187, 0.185, 0.201, 0.203, 0.255, 0.337$ 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 0.34 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.

Sensitivity to calibration. The three-run-mean calibration gives a sampled 344M peak $\mathrm{CEL} = 0.337$ and a smoothed-peak estimate 0.333. Omitting one model-size baseline mean at a time gives ranges $[0.228, 0.346]$ and $[0.223, 0.342]$, respectively. These describe calibration sensitivity, not seed-level confidence intervals. The absolute compute-equivalent values depend on the fitted reference; Appendix I reports the replication and uncertainty analyses.

4.4 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_{in} and x_{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. We fit weighted linear regression to the observed features, weighting each distinct training example by its multiplicity.

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 $\text{Cov}(z \mid X_{\text{in}}) = \|\beta_{\text{out}}\|_2^2 \Sigma_r$ for $z = X_{\text{out}}\beta_{\text{out}}$, with the block-diagonal matrix

$$\Sigma_r \;=\; I_n \;\oplus\; \bigoplus_{i=1}^{d} \mathbf{1}_r \mathbf{1}_r^\top \;\in\; \mathbb{R}^{(n+rd)\times(n+rd)}, \tag{8}$$

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. Conditioning on X_{in}, the expected train and test errors are

$$\mathbb{E}[L_{\text{train}} \mid X_{\text{in}}] = \frac{\|\beta_{\text{out}}\|_2^2}{n+rd}\left[n + rd - \text{tr}\big((C_u + r^2 C_d)(C_u + rC_d)^{-1}\big)\right], \tag{9}$$

$$\mathbb{E}[L_{\text{test}} \mid X_{\text{in}}] = \|\beta_{\text{out}}\|_2^2\left[1 + \text{tr}\big((C_u + rC_d)^{-1}(C_u + r^2 C_d)(C_u + rC_d)^{-1}\big)\right]. \tag{10}$$

equation 9–equation 10 are closed-form conditional risks. The non-monotonic behavior is a simulation-supported mechanism explored in §4.5. Appendix D gives the derivation.

What the formulas suggest. The observed dimension m represents model capacity and d the repeated-pool size. At fixed observed sample matrices, equation 10 is nondecreasing in $r \geq 1$; it does not predict recovery by increasing the weight of a fixed pool. Under a fixed total budget M and repeated fraction f, however, $n = (1-f)M$ and $d = fM/r$: increasing r reduces the number of distinct repeated examples. The simulations explore pool size and repetition jointly and provide a qualitative analogue, not a quantitative theory of transformer scaling.

4.5 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 equation 10 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 6 shows that the closed-form and simulation exhibit qualitatively consistent excess-loss surfaces, 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 $D_{\text{r}}^{\text{peak}}(N)$ relationship in equation 4.

We also compute a sample-efficiency analogue of CEG. For each repeated-data run, we estimate the no-repetition unique sample budget $N_{\text{clean}}^\star$ required to match its test loss, and define $\text{SE} = N_{\text{clean}}^\star / N_{\text{actual}}$, with the repeated block again accounting for 10% of the training budget. Figure 7 shows that SE falls sharply at intermediate r and partially recovers at extreme r, mirroring the CEG curve in Figure 5 for language models. Figure 8 shows

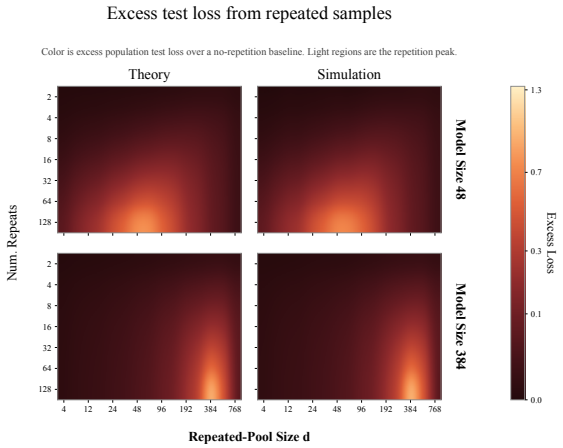

Figure 6: Excess test loss in misspecified linear regression with repeated samples. Rows vary the observed dimension m, and columns compare closed-form theory with direct OLS simulations. Within each panel, the x-axis is the repeated-pool size d, the y-axis is the repeat count r, and color indicates excess population test loss over a no-repetition baseline at the same m. The peak shifts to larger d as m increases, providing a statistical analogue consistent with the $D_r^{\text{peak}}(N)$ trend in Figure 3.

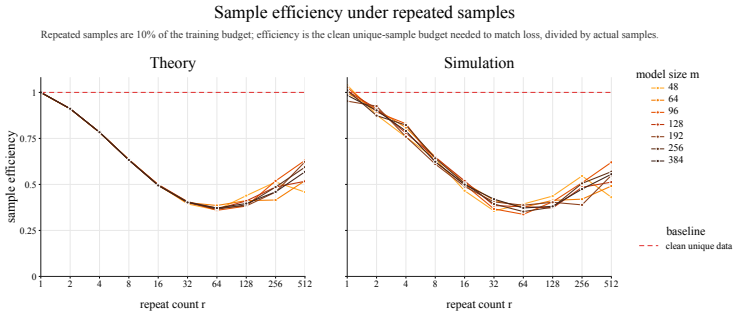

Figure 7: Sample efficiency under repeated samples. The repeated block accounts for 10% of the training budget. For each repeat count r, SE is the unique-sample budget needed to match the repeated-data test loss, divided by the actual sample budget. Theory and simulation both show non-monotonic efficiency loss, mirroring Figure 5.

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.

5 Conclusion

At a fixed repeated-token fraction, repetition structure matters: intermediate repeat counts produce the largest loss increases in our controlled sweeps. Larger models tend to peak at fewer repeats of larger pools within the measured range. At 344M and $\text{OT} = 1$, the calibration gives a smoothed-peak CEL of 0.333, with substantial calibration sensitivity. Misspecified linear regression provides a qualitative analogue when pool size and repetition vary jointly. These results motivate reporting both the amount and concentration of repeated training data.

AI Use Statement

In this work, we used generative AI tools for feedback on training configurations and manuscript analyses. We developed the original theory independently and subsequently used AI assistance to review its

presentation and interpretation. Additionally, we used AI agents to rerun analysis code, recompute fits, generate figures, and help debug training issues, including OOM errors and batch-size adjustments, and to improve the grammar and conciseness of drafts initially written by the authors. We reviewed all AI-assisted work, reread the revised text, and verified the analyses. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility Statement

Model configurations are provided in Appendix B, and data splitting and repeated-pool construction are described in Appendix C. Training, evaluation, and fitting procedures are described in Appendix F. The derivation of the linear-regression risk formulas is provided in Appendix D.

References

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.

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.

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.

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.

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.

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.

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.

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.

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*, pp. 4345–4382, 2024. URL https://proceedings.mlr.press/v235/bordelon24a.html.

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.

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)*, pp. 2633–2650, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

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.

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.

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*, pp. 16078–16092, 2024. URL https://aclanthology.org/2024.findings-acl.951/.

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.

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.

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.

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.

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.

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.

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.

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), pp. 5075–5084, 2023. URL https://aclanthology.org/2023.emnlp-main.308/.

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*, pp. 10697–10707, 2022. URL https://proceedings.mlr.press/v162/kandpal22a.html.

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.

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.

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

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/.

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)*, pp. 15616–15635, 2024. URL https://aclanthology.org/2024.acl-long.834/.

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.

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*, pp. 3245–3276, 2024. URL https://aclanthology.org/2024.naacl-long.179/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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.

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, pp. 157–165, 2022. URL https://aclanthology.org/2022.acl-short.18/.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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)*, pp. 15725–15788, 2024. URL https://aclanthology.org/2024.acl-long.840/.

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.

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.

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.

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)*, pp. 38274–38290, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/fa0509f4dab6807e2cb465715bf2d249-Abstract-Conference.html.

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.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017. URL https://arxiv.org/abs/1706.03762.

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.

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.

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.

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.

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.

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*, 2025a. URL https://arxiv.org/abs/2505.09388.

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

APPENDIX

A COMPREHENSIVE RELATED WORK

We expand here the discussion sketched in §2, organized along five threads.

Repeated data in language model pretraining. Our closest predecessor is Hernandez et al. (2022), 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 degradation of induction heads (Hernandez et al., 2022). The complementary regime, in which the entire corpus is uniformly repeated, is studied by Muennighoff et al. (2023). They find that up to roughly four epochs can be nearly as useful as fresh data in their tested compute-constrained setting, and model diminishing returns through effective data and parameter counts. Xue et al. (2023) study multi-epoch training and show that its benefits and degradation depend on the data, model, training objective, and regularization; they do not establish a universal four-epoch threshold. Other work in this family includes Komatsuzaki (2019), who studies one-pass-vs-multi-pass tradeoffs at smaller scale, and Maini et al. (2024), who replaces literal repeats with paraphrases generated by a teacher model. Tirumala et al. (2022) study memorization and forgetting during training, while Lesci et al. (2024) develop a causal account of memorization. Kazdan et al. (2026) argue that semantic duplication is itself scale-dependent, in the sense that larger models recognize more documents as duplicates. Lovelace et al. (2026) fit a one-parameter overfitting penalty within Chinchilla scaling that is closely related in spirit to our $\mathrm{CEG}/\mathrm{CEL}$ metric. Like Hernandez et al. (2022), we study a fixed minority fraction $f = 0.1$ of repeated training tokens, complementing the uniform-repetition setting of Muennighoff et al. (2023). We extend this line of work by measuring damage in compute-equivalent units relative to a fitted no-repetition scaling law and fitting the most damaging repeated-pool size across our model-size-dependent token budgets. These fitted peak locations differ from the boundaries of the harmful-repetition region studied by Hernandez et al. (2022).

Compute-optimal scaling. The power-law form for transformer loss was established by Kaplan et al. (2020) and refined by Hoffmann et al. (2022), 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. (2024) reanalyze the original Chinchilla fits, Porian et al. (2024) reconcile competing exponents across studies, Sardana et al. (2024) extend the framework to inference-aware budgets, and Gadre et al. (2025) show that Chinchilla-style fits extrapolate reliably under aggressive over-training in the regime we use. Caballero et al. (2023); Bahri et al. (2024); Bordelon et al. (2024); Henighan et al. (2020) provide alternative parametric forms and theoretical accounts. Tay et al. (2022) 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. (2022), who show that exact and near-duplicate removal improves training efficiency and reduces verbatim regurgitation. Kandpal et al. (2022) document a superlinear amplification of privacy risk by duplicates, and Carlini et al. (2023; 2021) relate memorization to model scale and duplicate count. Semantic deduplication was advanced by Abbas et al. (2023) and Tirumala et al. (2023), who use embedding clusters to remove near-duplicate documents that elude exact-hashing pipelines. Aggregate surveys include Deng et al. (2024) and the data-selection survey of Albalak et al. (2024). A related but distinct concern is benchmark contamination, where evaluation data leaks into training (Deng et al., 2024; Schaeffer et al., 2026; Oren et al., 2024; Magar & Schwartz, 2022; Jacovi et al., 2023). Schaeffer et al. (2026) show that even a single test-set replica can drive measured loss below an estimated uncontaminated loss floor fitted in their MATH experiments. 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 (Penedo et al., 2024). DataComp-LM provides a testbed for data curation (Li et al., 2024); Dolma is a mixed-source pretraining corpus (Soldaini et al., 2024); RedPajama-v2 provides web data with quality signals (Weber et al., 2024); and RefinedWeb is a filtered and deduplicated web corpus (Penedo et al.,

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

Model	Layers	d_{model}	d_{ff}	Heads	KV heads	d_{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

2023). Longpre et al. (2024) systematically investigate pretraining-data design choices. On the data-selection side, Sorscher et al. (2022) show that careful pruning can break power-law scaling, Marion et al. (2023) study perplexity-based pruning, Xie et al. (2023) optimize domain mixtures, and Goyal et al. (2024) 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. We use a fitted no-repetition scaling law as a common reference for quantifying our repetition interventions.

Statistical accounts of overfitting. The closed-form analysis in §4.4 sits in the literature on benign overfitting and double descent (Belkin et al., 2019; Nakkiran et al., 2020; Hastie et al., 2022; Bartlett et al., 2020; Belkin et al., 2020; Advani et al., 2020; Tsigler & Bartlett, 2023; Mei & Montanari, 2022), 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 equation 8 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.

B ARCHITECTURE.

We instantiate all models from scratch using the Qwen3 decoder architecture family (Yang et al., 2025a). 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 (Su et al., 2024), RMSNorm, SwiGLU feed-forward layers, multi-head attention with equal query and key-value head counts, 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.

Compute accounting. We use $C = 6NT$ as a parameter-based compute proxy, with N including both the input embedding matrix and the untied output head. These components have different operation costs, and the proxy does not separately account for sequence-length-dependent attention operations. Our configurations have large embedding fractions and an attention width of $32 \times 128 = 4096$, independently of the hidden width. Consequently, CEG and CEL quantify equivalence under this common proxy and fitted reference; they are not validated measurements of physical FLOP or wall-clock savings across architectures.

C REPEATED-POOL CONSTRUCTION DETAILS

For the original scaling grid, 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 Yang et al. (2025a) 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

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

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 $D_{\mathrm{r}}^{\star}$. 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 $\widehat{D}_{\mathrm{r}}$ and realized repeated-token fraction $\widehat{f}$ are approximate. If $L_{\max}$ is the maximum tokenized document length after truncation and EOS insertion, then

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

The non-repeated pool has the same one-document overshoot bound. In our preprocessing, $L_{\max} \le 2049$ 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, $D_{\mathrm{r}} = fT/R$ denotes the target unique repeated-pool size, so $fT = RD_{\mathrm{r}}$ is an exact identity for the targets. Rounding instead affects $R\widehat{D}_{\mathrm{r}}$. If $\widehat{U}$ is the realized non-repeated token count, the construction above gives

$$T_{\mathrm{constructed}} = \widehat{U} + R\widehat{D}_{\mathrm{r}}, \qquad T \le T_{\mathrm{constructed}} < T + (R+1)L_{\max},$$

with realized repeated fraction $\widehat{f} = R\widehat{D}_{\mathrm{r}}/T_{\mathrm{constructed}}$. Thus a one-document overshoot in the repeated pool is amplified by R. In the exported results, the 48M, OT $= 0.25$, $R = 10000$ run records 252,860,630 tokens against a nominal budget of 240,023,680, a 5.35% excess. Recorded dataset-token counts do not by themselves establish the exact number processed by optimizer steps. The peak-scaling fits use nominal budgets; their sensitivity to realized token counts and repeated fractions has not yet been established. The additional replication in Appendix I varies the split seed as well as the training and shuffle seeds.

D Linear regression with repeated samples

We derive the formulas used in Section 4.4. 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_{in} and the unobserved-feature matrix as X_{out}. Labels are noiseless:

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

We assume that the matrix of the $n+d$ distinct observed examples has full column rank m, and that $r \ge 1$. Repetition does not increase rank. For independent Gaussian examples this requires $n + d \ge m$ and holds almost surely under that condition; $n + rd \ge m$ alone is insufficient. The inverse formulas below apply on this full-rank domain, not to an underdetermined pseudoinverse solution. If risk is also averaged over the Gaussian observed design and the omitted signal is nonzero, a finite unconditional mean requires $n + d > m + 1$.

The restricted OLS estimator is

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

where

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

Thus a_r captures aliasing from fitting the unobserved signal with observed features.

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_{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,

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

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

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

we obtain

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

For test loss, take a fresh example $(x_{\text{in}}, x_{\text{out}}) \sim \mathcal{N}(0, I_p)$ from the population, with x_{in} and x_{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_{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

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

and therefore

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

E THEORY: REPETITION PEAK HEATMAP

The expression in Appendix D is an expectation over the omitted training coordinates conditional on the observed design. A direct OLS fit on one realized training set has a random population risk; averaging independent training realizations approximates the conditional expression with Monte Carlo error. This is distinct from the numerical equivalence of expanded and multiplicity-weighted OLS on identical data. The figures illustrate qualitative agreement. Grid resolution limits the precision of a sampled peak location and is not a Monte Carlo confidence interval.

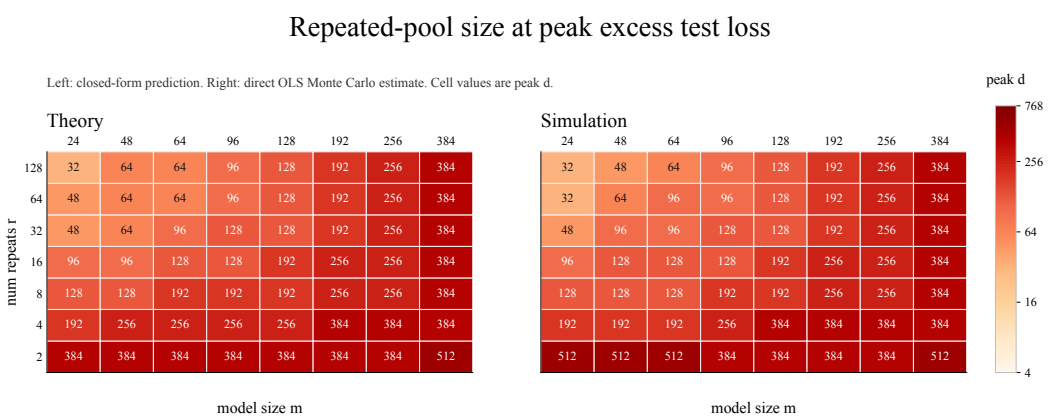

Figure 8: To isolate the peak location, we fix (m, r) and report the repeated-pool size d that maximizes excess test loss. The same trend appears in both the closed-form risk and direct OLS simulations: larger-capacity models peak at larger repeated pools, while higher repeat counts shift the peak toward smaller pools. Theory and simulation have qualitatively similar peak locations, sometimes differing by a neighboring value on the d grid.

F Training and Evaluation Details

Architecture. We use Qwen3-style decoder-only transformers (Yang et al., 2025a; 2024; 2025b) with rotary position embeddings (Su et al., 2024), RMSNorm, SwiGLU feed-forward layers, equal query and key-value head counts, 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 use $C = 6NT$ as a compute proxy based on the standard dense-transformer estimate (Kaplan et al., 2020; Hoffmann et al., 2022), subject to the architecture-specific qualifications in Appendix B.

Optimizer and schedule. All runs use AdamW Loshchilov & Hutter (2019) 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.

Data and evaluation. The source corpus is FineWeb-Edu-Dedup from the HuggingFaceTB SmolLM corpus Penedo et al. (2024). For the original scaling grid, 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 Yang et al. (2025a), 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 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, 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 (N, OT, R) point in this original grid is a single training run; the additional replications are reported separately.

Peak fitting. For each completed (N, 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

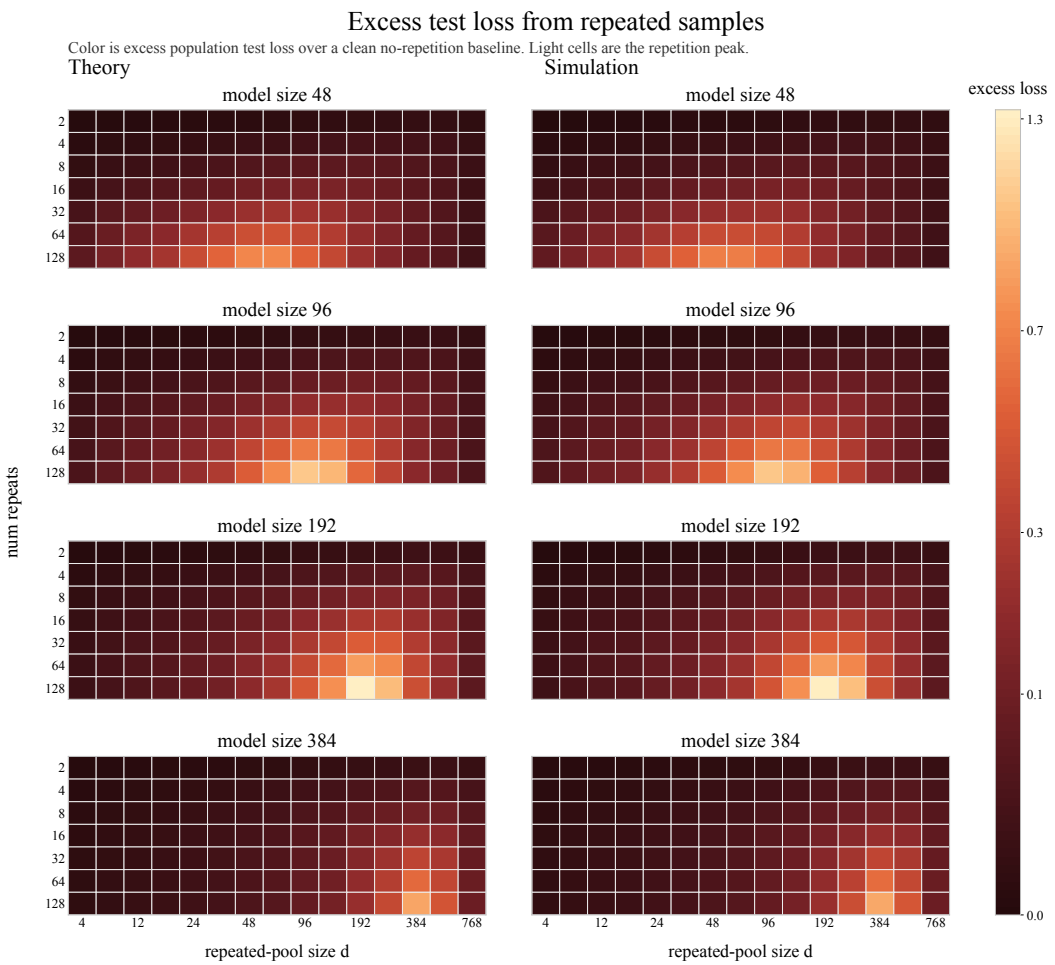

Figure 9: Full visualization of Figure 6

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 $R^{\text{peak}} = \mu$, and the corresponding repeated-pool size is computed from $D_r^{\text{peak}} = 2\text{OT}N/R^{\text{peak}}$. Power-law fits in equation 4–equation 5 are ordinary least-squares regressions in log-log space over the completed sweeps.

Scaling-law fitting. We fit $L(C) = E + KC^{-\gamma}$ to the mean loss of three no-repetition OT=1 runs at each of six model sizes, using nominal $C = 120N^2$ and the exact total parameter counts in Table 1. Nonlinear least squares minimizes residuals in $\log_{10} L$, with initial K, γ from a log–log fit without an offset and bounds $0 \le E < \min L$, $10^{-5} \le K \le 10^{20}$, $0.001 \le \gamma \le 2$. Original per-sweep $R = 1$ losses remain the anchors for peak-shape fitting. Recorded token counts are used for the compute proxy of sampled repeated-data runs; smoothed curves use nominal budgets. The exported counts have not been verified as exact optimizer-processed token totals (Appendix C). Appendix I reports sensitivity to calibration and peak fitting.

Reported evaluation loss. All losses used in the original scaling grid are final-checkpoint losses. After each run reaches its target token budget $T = 20 \cdot \text{OT} \cdot N$, 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.

G INTERPRETING CEG > 1

The CEG ratio in equation 7 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 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 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 Yang et al. (2025a) models on FineWeb-Edu-Dedup Penedo et al. (2024) 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, 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.

H LIMITATIONS

Our largest model has 344M parameters and all experiments use one Qwen3-style architecture family and tokenizer. The main $f = 0.1$ scaling grid uses FineWeb-Edu-Dedup and predominantly single runs. The additional baseline replications, one 93M seed/shuffle/split replication, and $f = 0.05$ sweeps broaden the evidence but do not establish seed-level confidence intervals for every peak or general scaling across corpora and repetition fractions. The two 93M seed settings have substantially different sampled peak locations. Compute constraints left the larger-model grid incomplete: 153M runs through OT $= 2$ and 344M only at OT $= 1$. We report completed runs and do not impute losses for uncompleted configurations. The reported empirical outcomes are held-out language-modeling losses; we do not establish effects on downstream task accuracy or generation quality. The absolute CEG and CEL values depend on a three-parameter fit to six baseline means and on the nominal-budget convention for these means. We do not validate extrapolation beyond the measured scales. The linear model omits attention, depth, optimization dynamics, and discrete tokens, and supplies a qualitative statistical analogue.

I ADDITIONAL EMPIRICAL RESULTS AND SENSITIVITY ANALYSES

I.1 CALIBRATION AND PEAK-FITTING PROCEDURE

Displayed estimates are rounded to three significant figures; all calculations use full-precision values. Differences and standard deviations are computed before rounding. We refit the original $f = 0.1$ results using the exact total parameter counts in Table 1. The no-repetition calibration uses the six mean losses from three $R = 1$, OT $= 1$ runs per model size, fitting $L(C) = E + KC^{-\gamma}$ by least squares in $\log_{10} L$, with $C = 120N^2$ computed from nominal $T = 20N$ budgets. The fitted values are $E = 2.36$, $K = 6.45 \times 10^5$, and $\gamma = 0.316$ when C is expressed in the FLOP units of the $6NT$ proxy, with the limitations described in Appendix B. The baseline means are reported below. Actual token counts for these replications are unavailable in the supplied table; we use their confirmed OT=1 nominal budgets. Original per-sweep clean baselines remain the references for peak fitting. For each (N, OT) cell, we fit

$$\eta(R) = A \exp\left[-\frac{(\log_{10} R - \log_{10} \mu_R)^2}{2\sigma^2}\right], \qquad \eta(R) = \frac{L(R) - L(1)}{L(1)}. \tag{11}$$

The $R = 1$ anchor is excluded from this fit, and at least four nonbaseline measurements are required. This yields 25 fitted cells. We use nominal budgets $T = 20\,\mathrm{OT}\,N$ and $C = 120\,\mathrm{OT}\,N^2$ for peak-scaling fits, with $D_r^{\mathrm{peak}} = 0.1T/\mu_R$. The four power laws are separate ordinary least-squares fits in log–log coordinates pooled across these cells. Consequently, their slopes need not obey transformations that would apply to a single fixed-OT scaling relation.

I.2 Fit sensitivity

We perform 10,000 paired-cell bootstrap resamples and report the 2.5th and 97.5th percentiles. A second analysis resamples the six model-size groups, keeping all available OT cells within each selected group. Both analyses condition on the fitted cell peaks: neither propagates all uncertainty in each peak fit nor measures training-seed variability. Resampling six model-size groups also provides limited information about extrapolation.

Table 2: Conditional bootstrap intervals for the four peak-scaling slopes.

Fit	Slope	Cell bootstrap	Size-group bootstrap
R^{peak} vs. N	-0.872	[-0.962, -0.773]	[-0.884, -0.862]
D_r^{peak} vs. N	1.75	[1.18, 2.25]	[1.55, 1.88]
R^{peak} vs. C	-0.217	[-0.330, -0.0830]	[-0.316, -0.00252]
D_r^{peak} vs. C	0.905	[0.872, 0.936]	[0.866, 0.931]

For R^{peak} versus N, deleting one cell produces slope estimates ranging from -0.900 to -0.849. The approximate normal jackknife interval is $[-0.963, -0.782]$; this is distinct from the range of delete-one estimates. The 344M sampled-peak CEL is 0.337, with leave-one-baseline-out range $[0.228, 0.346]$. The smoothed-peak CEL is 0.333, with range $[0.223, 0.342]$. These ranges describe calibration sensitivity, not seed-level confidence intervals. The sampled estimate uses $6N$ times the recorded token count as its compute proxy; the smoothed estimate uses the nominal OT budget. Neither is a hardware-profiled FLOP measurement. These sensitivity analyses do not assess the token-rounding effects discussed in Appendix C.

I.3 No-repetition baseline replication

We additionally report three no-repetition runs per model size. Means and sample standard deviations are recomputed from the reported losses, with denominator $3 - 1$ for sample variance. These measure baseline variability, not variation in the repeated-data peak.

Table 3: No-repetition losses across three runs.

Model	Run 1	Run 2	Run 3	Mean	Sample SD
34M	4.87	4.86	4.86	4.86	0.00793
48M	4.34	4.35	4.34	4.34	0.00256
63M	4.05	4.05	4.05	4.05	0.00100
93M	3.69	3.68	3.68	3.68	0.00437
153M	3.33	3.33	3.33	3.33	0.00107
344M	2.93	2.93	2.93	2.93	0.000575

I.4 Repeated-data replication at 93M

The additional $f = 0.1$, $\mathrm{OT} = 1$ experiment changes the training seed, shuffle seed, and train/test split seed together. It therefore probes combined run variability rather than isolating initialization variance. Multiple records at the same (seed, R) are averaged before comparison; each seed setting uses its own $R = 1$ reference. The sampled peaks occur at $R = 193$ for seed 0 and $R = 2682$ for seed 1, with relative loss increases of 2.74% and 1.39%, respectively. Compute constraints limited the seed-1 sweep to a sparser repeat-count grid, so its sampled maximum may miss a higher peak at untested repeat counts. Both settings include both reported peak repeat counts, so different grid coverage alone does

not explain the observed disagreement. Both show lower loss at $R = 10000$ than at their respective sampled maxima, supporting the qualitative pattern while leaving the peak location uncertain.

Table 4: 93M replication. Compute constraints limited the seed-1 grid. Dashes indicate unavailable cells.

R	Seed 0 loss	Seed 1 loss	Seed 0 Δ (%)	Seed 1 Δ (%)
1	3.67	3.69	0.00	0.00
3	3.67	–	0.0161	–
13	3.68	3.69	0.170	0.143
51	3.70	–	0.620	–
99	3.74	–	1.81	–
193	3.77	3.69	2.74	0.137
373	3.77	–	2.62	–
719	3.77	3.69	2.60	0.0864
2682	3.73	3.74	1.52	1.39
10000	3.71	3.71	0.955	0.801

I.5 A SMALLER REPEATED-TOKEN FRACTION

At $f = 0.05$ and $\mathrm{OT} = 1$, we evaluate 34M, 63M, 93M, and 153M models. Several curves have an intermediate-repeat sampled maximum, while the 153M curve is nearly flat. The 63M sweep lacks a completed $R = 1$ point, so its reported increase is relative to $R = 13$, not a no-repetition baseline. These follow-up sweeps were compute-constrained and sparsely sampled, so their sampled maxima may miss higher peaks at untested repeat counts. Denser sweeps are needed to resolve peak locations and heights, including for the nearly flat 153M curve. These sparse sweeps do not establish a scaling law in f.

Table 5: Held-out losses at $f = 0.05$ and $\mathrm{OT} = 1$.

Model	1	13	50	80	193	719	1000	2682	5000	10000
34M	4.86	4.86	–	–	4.88	4.86	–	4.92	4.86	4.86
63M	–	4.07	4.07	–	4.11	–	4.13	4.11	–	4.09
93M	3.68	3.69	3.68	3.70	3.73	3.68	3.72	3.71	–	3.70
153M	3.33	3.33	–	–	3.33	–	–	3.33	–	3.32

Table 6: Sampled increases from full-precision losses; the 63M reference contains repetition.

Model	Reference R	Peak R	ΔL	$\Delta L / L_{\mathrm{ref}}$ (%)
34M	1	2682	0.0640	1.32
63M	13	1000	0.0589	1.45
93M	1	193	0.0468	1.27
153M	1	13	0.00376	0.113

I.6 EXTENSION FIGURES

The following plots reuse the original analysis implementation. Dashed curves are descriptive log-Gaussian fits, not confidence bands. In particular, smooth curves through sparse measurements should not be taken as evidence for precisely located peaks.

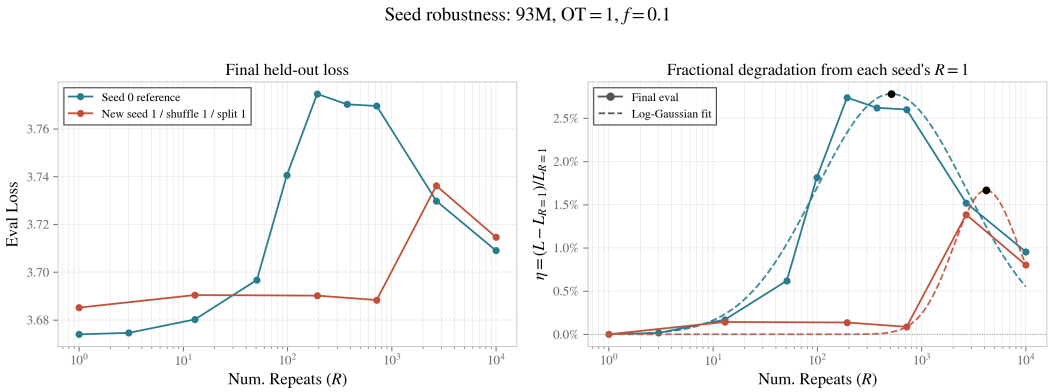

Figure 10: 93M replication at $f = 0.1$ and $\mathrm{OT} = 1$. Seed, shuffle, and split vary together. Compute constraints limited the seed-1 grid; its sampled maximum may miss a higher peak at untested repeat counts.

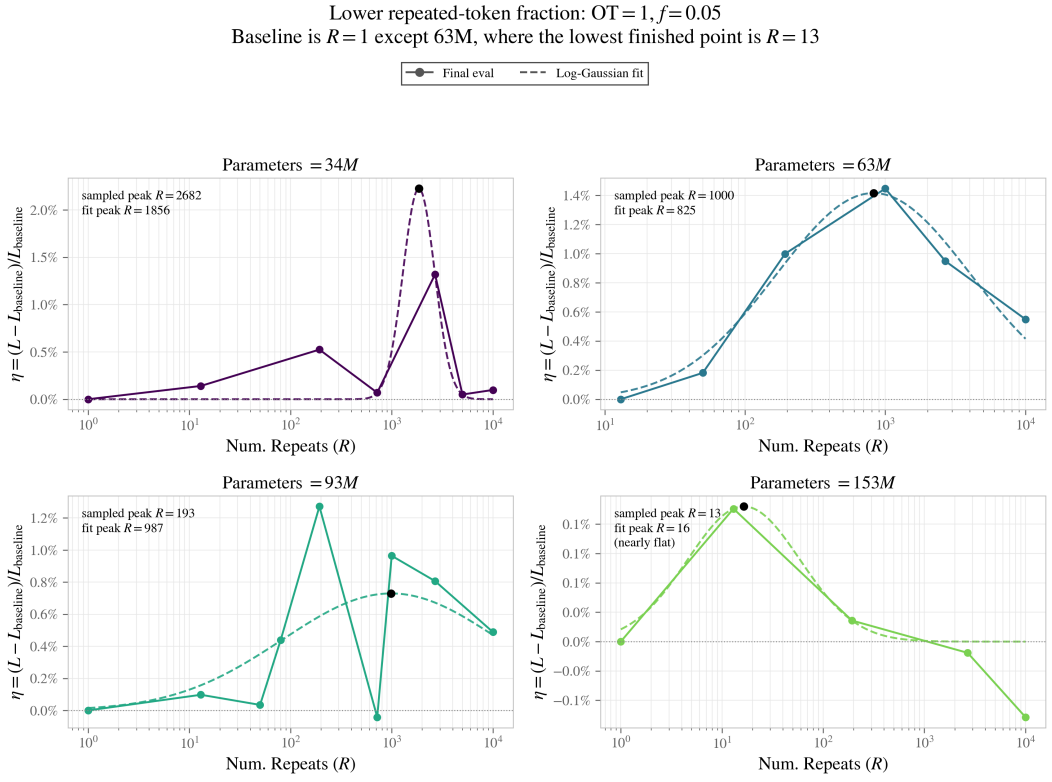

Figure 11: Smaller repetition-fraction sweeps. The 63M reference is $R = 13$; all other references use $R = 1$. These compute-constrained grids may miss higher peaks at untested repeat counts.