SCALING DOMAIN DATA REPETITION IN LLM PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models scale, their training-token budgets must also increase to maintain an appropriate tokens-per-parameter ratio (TPP). However, high-quality domain data is much harder to scale than general web data. As model size and the training-token budget increase, its fraction in the training mixture tends to decrease. Repeating the available high-quality data provides an effective way to counteract this dilution, but excessive repetition may lead to overfitting. We study this trade-off under practical LLM scaling, where the training-token budget grows proportionally with model size. For a fixed domain, we first find that, surprisingly at a fixed TPP, the optimal repetition count mildly increases with model size. Across different domains, we find that the optimal repetition count is strongly negatively correlated with the final validation loss of a domain: domains with lower loss can generally benefit from more repetitions. In contrast, the amount of unique domain data is only weakly related to the optimal repetition count. These findings suggest that repetition counts tuned on smaller proxy models with the same TPP can provide a practical estimate for larger models.

1 INTRODUCTION

Large language models (LLMs) are often pretrained on mixtures of diverse domains, including general web text, code, mathematics, scientific corpora, and multilingual content (Doddapaneni et al., 2025; Taylor et al., 2022; Team et al., 2023; Dubey et al., 2024). As model size increases, compute-optimal training also requires an increasing number of training tokens (Kaplan et al., 2020; Hoffmann et al., 2022). This creates an asymmetric data-scaling problem: broad general web data (Raffel et al., 2020; Penedo et al., 2024; Li et al., 2024) can often be scaled relatively easily, whereas high-quality domain-specific data (Paster et al., 2024; Li et al., 2023; Guo et al., 2020) is much harder to scale at the same rate. However, if the amount of such data remains fixed, its mixing ratio decreases as the total training budget grows. Recent work shows that under data mixing, knowledge-dense domains may only be learned once their mixing ratio exceeds a critical threshold; below this threshold, additional training can still fail to acquire the target knowledge (Gu et al., 2025). Therefore, maintaining a sufficient fraction of high-quality domain data is important when scaling data.

Prior work has extensively studied data repetition, where examples from a finite dataset are reused multiple times during training (Muennighoff et al., 2023; Lovelace et al., 2026; Sedova et al., 2026; Wu et al., 2026). For high-quality domain data, repetition offers a direct way to counteract data dilution, but may also increase the risk of overfitting (Lee et al., 2022; Carlini et al., 2022; Hernandez et al., 2022; Xue et al., 2023). We therefore need to study the trade-off between the domain-learning gains from repeated high-quality data and the risk of overfitting, and to identify the optimal number of repetitions for different domains.

A first line of work studies data repetition in the single-dataset setting (Muennighoff et al., 2023; Xue et al., 2023; Lovelace et al., 2026). More recent work extends this question to mixtures of general and domain-specific data, showing that general data can regularize repeated domain data and developing scaling laws that account for repetition under data mixing (Sedova et al., 2026; Liu et al., 2026; Chudnovsky et al., 2026). Across these settings, the conclusion is that larger models are more susceptible to overfitting from repeated data, suggesting that repetition should be minimized when training large LLMs. We confirm this with our experiments shown on the left in Figure 1.

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.

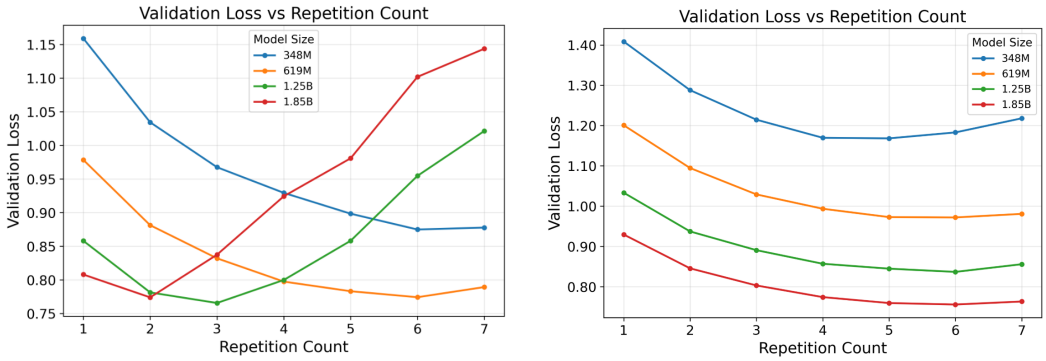


Figure 1: **Data repetition under two model-scaling setups. Left: Fixed total training tokens across model sizes.** When the training-data size is fixed independently of model size, the optimal repetition count decreases with model size. **Right: Fixed TPP ($D = cN$) across model sizes.** When the tokens-per-parameter ratio TPP is fixed, the optimal repetition count increases with model size. The fixed-TPP setting better reflects practical LLM scaling and is the focus of our study.

However, we also note that existing cross-scale comparisons typically hold the training-data size fixed rather than preserving the tokens-per-parameter ratio (Muennighoff et al., 2023; Xue et al., 2023; Chudnovsky et al., 2026), defined as $\text{TPP} := D/N$, where D is the total number of training tokens and N is the model size. This distinction leads to different conclusions about repetition. As illustrated in Figure 1, when D is fixed, increasing the model size exposes a larger model to the same repeated dataset, causing overfitting to occur earlier and the optimal repetition count to approach one. *In contrast, when the training-token budget grows with model size using a fixed* TPP (Hoffmann et al., 2022; Kalra & Barkeshli, 2026), *the optimal repetition count surprisingly increases with model size.*

To study repetition under this scaling regime, we consider four high-quality domains: code, math, Wikipedia and medical data. For each domain, we vary its amount of unique data and repetition count while keeping the total training-token budget fixed for each model size. This setup allows us to isolate how data repetition interacts with domain properties, unique-data size, and model scale.

Our main findings are as follows.

- **The optimal repetition count is strongly negatively correlated with the final validation loss of a domain.** Domains with lower validation loss can generally tolerate and benefit from more repetitions, whereas domains with higher validation loss overfit earlier.
- **At a fixed tokens-per-parameter ratio** TPP**, the optimal repetition count increases with model size.** This trend is opposite to that observed when the data budget is fixed across model sizes. It also supports conservative transfer from smaller proxy models: a repetition count that does not cause overfitting in a smaller model is unlikely to cause overfitting in a larger model trained at the same TPP.
- **The optimal repetition count is largely insensitive to the amount of unique data.** Across the tested fractions of unique high-quality tokens, the repetition count that minimizes validation loss remains nearly unchanged. Therefore, when estimating the optimal repetition count on a smaller proxy model, it is not necessary to use a specific unique high-quality token fraction; any representative fraction within the tested range can be used.

Together, these results provide a practical approach for configuring the optimal repetition counts for high-quality domains. Repetition counts can first be swept on a smaller proxy model with the same TPP for an arbitrary fraction of unique high-quality tokens, and the result can be used to select relatively safe repetition counts for the target model, since we find that *a repetition count that does not cause overfitting on the proxy model is also safe for a larger model under the same TPP*.



Figure 2: **Overview of our experimental framework.** We train models of different sizes at a fixed tokens-per-parameter ratio and vary the amount and repetition count of high-quality domain data in the training mixture. We evaluate both in-domain validation loss and out-of-domain performance.

2 Setup

We study data repetition under different model sizes. Let $\mathcal{N} = \{N_1, \ldots, N_K\}$ denote the model sizes. For a model of size $N \in \mathcal{N}$, we set its total training-token budget to

$$D_N = \mathrm{TPP} \cdot N,$$

where TPP denotes the number of training tokens per parameter and is set to a constant greater than 100 in our experiments. Therefore, all training runs with the same model size use exactly the same number of training tokens, while larger models receive proportionally larger token budgets. We consider four high-quality domains. Let

$$\mathcal{D}_{\mathrm{HQ}} = \{\textsc{Code}, \textsc{Math}, \textsc{Wiki}, \textsc{Medical}\}$$

denote the set of high-quality domains. For each high-quality domain $d \in \mathcal{D}_{\mathrm{HQ}}$, we sweep over different combinations of unique-data fractions and repetition counts.

Let $\mathcal{A} = \left\{\frac{1}{40}, \frac{1}{20}, \frac{1}{10}\right\}$ denote the fraction of unique high-quality tokens. We set the number of unique high-quality tokens $U_{N,\alpha}$ as

$$U_{N,\alpha} = \alpha D_N, \quad \alpha \in \mathcal{A},$$

and the repetition count as $e \in \mathcal{E} = \{1, 2, 3, 4, 5, 6, 7\}$. In other words, for a configuration (d, N, α, e), we first select a fixed subset from domain d containing $U_{N,\alpha}$ unique tokens. This subset is then repeated e times. Hence, the total number of high-quality tokens during training is

$$H_{N,\alpha,e} = eU_{N,\alpha} = e\alpha D_N.$$

The remaining token budget is filled with non-repeated web data, whose token count is

$$W_{N,\alpha,e} = D_N - H_{N,\alpha,e} = (1 - e\alpha)\, D_N.$$

All configurations satisfy $e\alpha \leq 1$, and thus $W_{N,\alpha,e} \geq 0$. Importantly, α controls the amount of unique high-quality data, whereas e controls how often this fixed subset is revisited. Their product $e\alpha$ determines the final proportion of high-quality token presentations in the training stream. The total budget D_N remains unchanged across recipes for a fixed N; only its composition between the target high-quality domain and the web corpus changes. We do not mix multiple high-quality domains in the same run.

Let $\theta_{d,N,\alpha,e}$ denote the model obtained from configuration (d, N, α, e). The complete experiment grid for each domain $\mathcal{G}_d$ and the full set of experiments $\mathcal{G}$ are

$$\mathcal{G}_d = \mathcal{N} \times \mathcal{A} \times \mathcal{E}, \quad \mathcal{G} = \bigcup_{d \in \mathcal{D}_{\mathrm{HQ}}} (\{d\} \times \mathcal{G}_d)\,.$$

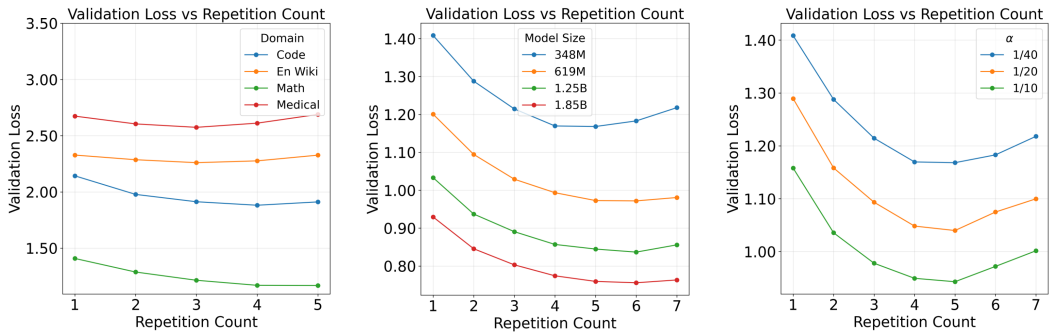


Figure 3: **Factors affecting the optimal repetition count. Left: Different high-quality domains.** Different high-quality domains exhibit substantially different optimal repetition counts. **Middle: Math domain across model sizes**. For the Math domain, the optimal repetition count increases with model size under a fixed TPP. **Right: Math domain across unique data fractions**. For the Math domain, varying α mainly shifts the validation loss but has little effect on the optimal repetition count. Overall, the domain has the strongest effect on the optimal repetition count, followed by model size, while α has little effect.

We evaluate each trained model using both in-distribution and out-of-distribution validation losses. For a validation corpus $\mathcal{V}$, we define the token-averaged negative log-likelihood as

$$\mathcal{L}(\theta; \mathcal{V}) = -\frac{1}{M_{\mathcal{V}}} \sum_{\boldsymbol{x} \in \mathcal{V}} \sum_{t=1}^{|\boldsymbol{x}|} \log p_\theta \left(x_t \mid x_{<t} \right),$$

where $\boldsymbol{x} = (x_1, \ldots, x_{|\boldsymbol{x}|})$ is a token sequence and $M_{\mathcal{V}} = \sum_{\boldsymbol{x} \in \mathcal{V}} |\boldsymbol{x}|$ is the total number of validation tokens. For a run targeting domain d, the in-distribution metric is

$$\mathcal{L}^{(d)}_{\mathrm{IID}}(N, \alpha, e) = \mathcal{L}\left(\theta_{d,N,\alpha,e}; \mathcal{V}^{\mathrm{IID}}_d\right),$$

where $\mathcal{V}^{\mathrm{IID}}_d$ is the held-out validation set from the same high-quality domain. We additionally report

$$\mathcal{L}^{(d)}_{\mathrm{OOD}}(N, \alpha, e) = \mathcal{L}\left(\theta_{d,N,\alpha,e}; \mathcal{V}^{\mathrm{OOD}}\right),$$

where $\mathcal{V}^{\mathrm{OOD}}$ is a pretraining validation set used to measure performance outside the repeated target domain. Our setup is illustrated in Figure 2.

3 Predicting the Optimal Repetition Count

In this section, we summarize the empirical factors associated with the optimal repetition count and provide a theoretical explanation. Empirically, the optimal repetition count depends mildly on model size, is largely insensitive to the fraction of unique high-quality data, and is strongly negatively correlated with the minimum validation loss. We then use a theoretical model to explain these observations through the trade-off between knowledge acquisition and noise fitting.

3.1 Empirical Findings

Recall that for each model size N, we train with a token budget $D_N = \mathrm{TPP} \cdot N$ and record the validation loss at the end of training. We vary the fraction of unique high-quality data, denoted by α, and its repetition count, denoted by e, while keeping the total number of training tokens fixed. We study how the optimal repetition count depends on α, model size, and data domain.

Figure 3 compares the effects of data domain, model size, and the fraction of unique high-quality data α on the optimal repetition count. The left panel shows a strong domain dependence. Math reaches its minimum at around 5 repetitions, while Wiki, Code, and Medical reach their minima earlier, indicating that the optimal repetition count varies substantially across domains. The middle panel shows the effect of model size on the Math domain at a fixed α. As model size increases, the optimal repetition count shifts toward larger values, suggesting a moderate dependence on model

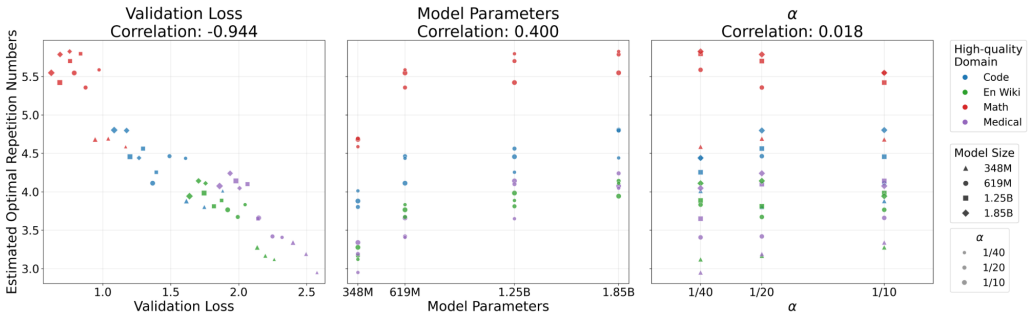


Figure 4: **The optimal repetition count is correlated with the high-quality domain loss, with a weaker dependence on model size and little observable dependence on α.** We estimate the optimal repetition count using the minimum of a quadratic fit to the final validation loss. From left to right, we show its relationship with the minimum validation loss, model size, and unique high-quality data fraction, with Pearson correlations of -0.944, 0.400, and 0.018, respectively. Colors denote domains, marker shapes denote model sizes, and marker sizes denote unique data fractions.

scale. The right panel shows the effect of α on the Math domain at a fixed model size. Although increasing α consistently reduces the validation loss, the location of the minimum remains nearly unchanged, indicating that α has little effect on the optimal repetition count. Overall, the optimal repetition count mainly depends on the data domain and model size, but is largely insensitive to α. The full results across all domains, model sizes, and values of α are provided in Appendix C.

To quantify the above observations, we estimate the optimal repetition count for each combination of domain d, model size N, and the fraction of unique high-quality data α, and then quantitatively characterize its dependence on these factors. Specifically, we fit a quadratic function to the final validation loss as a function of repetition count:

$$\widehat{L}_{d,N,\alpha}(e) = a_{d,N,\alpha}e^2 + b_{d,N,\alpha}e + c_{d,N,\alpha}.$$

We use the minimum of the fitted curve as the estimated optimal repetition count, $\widehat{e}^*_{d,N,\alpha} = -\frac{b_{d,N,\alpha}}{2a_{d,N,\alpha}}$. This continuous estimate reduces the effect of the discrete repetition grid, allowing clearer analysis of the correlations between the optimal repetition count and different factors.

Since different domains exhibit substantially different validation-loss levels, we further quantify the domain effect through validation loss. Specifically, we use the minimum validation loss in each setting as a continuous proxy for domain-specific characteristics and examine its relationship with the optimal repetition count.

Figure 4 compares the estimated optimal repetition count with the minimum validation loss, model size, and unique high-quality data fraction. The optimal repetition count is strongly negatively correlated with the minimum validation loss, with a Pearson correlation of -0.944: settings with higher validation loss generally prefer fewer repetitions. Model size shows a weaker positive correlation, indicating that larger models may support slightly more repetitions under a fixed tokens-per-parameter ratio. In contrast, the correlation with the unique high-quality data fraction is close to zero.

These findings suggest a practical procedure for selecting the repetition count for large-scale training. We first train a smaller proxy model with the same tokens-per-parameter ratio as the target model, using any suitable value of α. For each high-quality domain, its validation loss indicates a reasonable range of repetition counts to consider. The selected repetition count can then serve as a conservative estimate for the target model, since our experiments show that *a repetition count that does not cause overfitting on the proxy model is also safe for a larger model under the same TPP*.

3.2 Theoretical Analysis: Why Loss Determines Optimal Repetition

Our experiments reveal a strong negative correlation between validation loss and optimal repetition. We provide a simple theoretical explanation based on the following intuition: repetition initially helps the model learn the underlying signal, but excessive repetition eventually fits the noise in the finite training set. Cleaner data therefore tolerates more repetitions before overfitting.

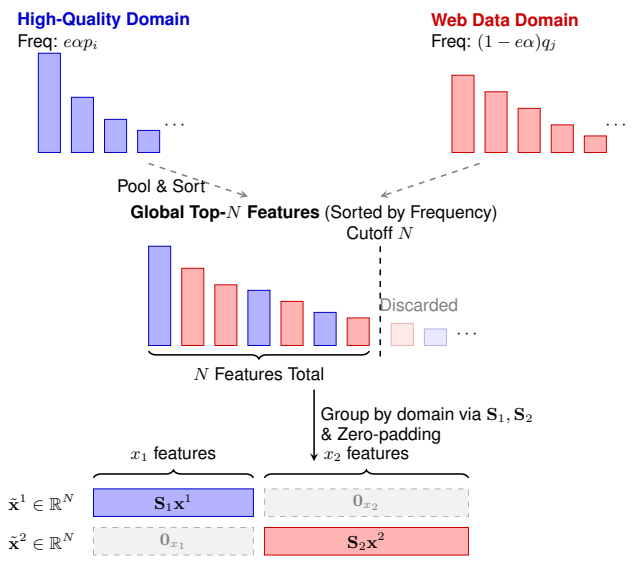


Figure 5: **Illustration of the feature selection and padding mechanism.** Features from both domains are pooled and globally sorted by unconditional frequencies. After restricting to the top-N features (x_1 from high-quality and x_2 from web), the observations are mapped to $\mathbb{R}^N$ via sketching matrices and zero-padding, ensuring orthogonal support for $\tilde{\mathbf{x}}^1$ and $\tilde{\mathbf{x}}^2$.

To model the limited capacity of the learner, we adopt a sketching-based formulation (Dai & Zheng, 2026; Li et al., 2025; Lin et al., 2024). Consider a linear regression problem with high-quality and web domains, whose ground-truth parameters are $\theta_1^*, \theta_2^* \in \mathbb{R}^{+\infty}$ and input covariates are $\mathbf{x}^1, \mathbf{x}^2$, respectively. Following Yan et al. (2025), we assume

$$\mathbb{P}(\mathbf{x}^1 = \mathbf{e}_i) = p_i = c_p i^{-s_1}, \qquad \mathbb{P}(\mathbf{x}^2 = \mathbf{e}_i) = q_i = c_q i^{-s_2},$$

where $s_1, s_2 > 1$. Since the high-quality and web domains occupy fractions $e\alpha$ and $1 - e\alpha$ of the training mixture, their features appear with unconditional frequencies $e\alpha p_i$ and $(1 - e\alpha)q_i$.

As illustrated in Figure 5, we use a simple top-N construction to model limited model capacity. We first pool all features from the high-quality and web domains and rank them by how frequently they appear in the training mixture. The model can represent only the N most frequent features. Among these N features, suppose x_1 come from the high-quality domain and x_2 come from the web domain, where $x_1 + x_2 = N$.

Let $\mathbf{S}_1$ and $\mathbf{S}_2$ extract these selected features from the two domains. We then place the high-quality features in the first x_1 coordinates and the web features in the remaining x_2 coordinates:

$$\tilde{\mathbf{x}}^1 = ((\mathbf{S}_1\mathbf{x}^1)^\top, \mathbf{0}_{x_2}^\top)^\top, \qquad \tilde{\mathbf{x}}^2 = (\mathbf{0}_{x_1}^\top, (\mathbf{S}_2\mathbf{x}^2)^\top)^\top.$$

Thus, the two domains share the same N-dimensional representation.

We assume that each label contains an underlying signal and additive noise:

$$y_i^1 = \langle \mathbf{x}_i^1, \theta_1^* \rangle + \varepsilon_i^1, \qquad y_i^2 = \langle \mathbf{x}_i^2, \theta_2^* \rangle + \varepsilon_i^2, \qquad \varepsilon_i^1, \varepsilon_i^2 \overset{iid}{\sim} \mathcal{N}(0, \sigma^2).$$

Since the amount of *unique* high-quality data is fixed and we have only

$$U_N = \alpha D_N = \alpha \,\mathrm{TPP} \cdot N$$

unique high-quality samples, increasing the repetition count e therefore means training longer on the same U_N samples, including their fixed sample-specific noise, rather than observing new high-quality data. This repeated exposure to the same noise is what eventually leads to overfitting.

Since the two domains occupy different coordinates after zero-padding, an SGD update from one domain does not modify the coordinates of the other domain. We can therefore analyze the high-quality domain independently. We focus on the high-quality coordinates $\theta \in \mathbb{R}^{x_1}$. Starting from

$\theta_0 = \mathbf{0}$, at each SGD step t, we uniformly sample an index

$$J_t \sim \mathrm{Unif}\{1, \dots, U_N\},$$

and update

$$\theta_{t+1} = \theta_t - \eta \nabla \ell_{J_t}(\theta_t), \qquad \ell_j(\theta) = \frac{1}{2}\big(\langle \mathbf{S}_1 \mathbf{x}_j^1, \theta\rangle - y_j^1\big)^2, \quad 0 \le t < \lfloor eU_N \rfloor.$$

Let $\mathbf{H} = \mathbf{E}[\mathbf{x}^1(\mathbf{x}^1)^\top]$. The population risk on the high-quality domain is

$$\mathcal{R}(\theta) = \frac{1}{2}(\mathbf{S}_1^\top \theta - \theta_1^*)^\top \mathbf{H}(\mathbf{S}_1^\top \theta - \theta_1^*) + \frac{1}{2}\sigma^2.$$

The noise variance σ^2 directly contributes to the irreducible component of the population risk. Therefore, for a fixed signal structure and model capacity, a smaller σ^2 leads to a lower attainable population loss. This provides a simple mechanism connecting the empirical observation that lower-loss domains tolerate more repetitions to the noise-dependent behavior analyzed below.

Following Lin et al. (2024), we impose the following source condition to characterize how the target signal is distributed across features:

Assumption 3.1 (Source condition)**.** *We assume that the coordinates of θ_1^* satisfy*

$$\mathbb{E}\big[p_i(\theta_{1,i}^*)^2\big] \asymp i^{-\beta}, \qquad \beta > 1.$$

This is a standard regularity assumption that controls how much signal lies in increasingly rare features. It is mainly used to make the risk decay analytically tractable; the key noise-repetition mechanism does not depend on its exact form.

Under this condition, the expected risk during SGD can be decomposed into two competing effects: a signal-fitting term that decreases with more training, and a noise-fitting term caused by repeatedly training on the same finite dataset. The latter is proportional to the noise variance σ^2. Therefore, when σ^2 is smaller, more repetitions are required before noise fitting dominates, leading to a larger optimal repetition count. The following theorem formalizes this relationship.

Theorem 3.2 (Small-noise repetition law)**.** *Suppose Assumption 3.1 holds with $\beta > 1$. Fix N, α, and $0 < \eta < 1$. After forming the top-N sketch, hold its nonempty high-quality support fixed and continuously extend the expected population risk to repetition counts $e \in [1, \infty)$. If $e_N^*(\sigma^2)$ is its smallest minimizer, then, as $\sigma^2 \downarrow 0$,*

$$e_N^*(\sigma^2) = \frac{\log(1/\sigma^2)}{U_N \log\left(\dfrac{1 - \eta/U_N}{1 - \eta(2-\eta)/U_N}\right)} + O(1).$$

In particular, $e_N^*(\sigma^2)$ is strictly decreasing in σ^2 for all sufficiently small σ^2.

Therefore, lower-noise domains admit a larger optimal repetition count, providing a theoretical explanation for the negative correlation between validation loss and optimal repetition observed in our experiments.

4 Additional Ablations and Analyses

In this section, we present additional analyses of data repetition. Section 4.1 compares repeated and unique high-quality tokens at a fixed total domain fraction and shows that the effect of repetition is strongly domain dependent. We also examine OOD pretraining performance and the results are shown in Appendix D. Section 4.2 examines whether the optimal repetition count depends on the learning-rate schedule.

4.1 Unique versus Repeated Tokens at a Fixed Domain Fraction

In the previous section, we fix the amount of unique high-quality data and vary its repetition count. We now consider the alternative setup from Muennighoff et al. (2023), where the total number of

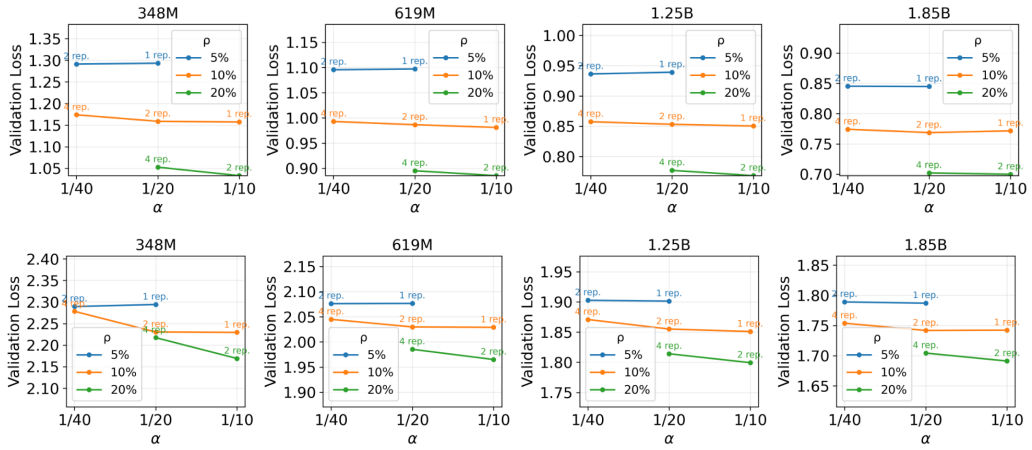


Figure 6: **Final validation loss at fixed total high-quality data fractions.** Top: Math. Bottom: Wiki. We fix the total high-quality-data fraction and vary the repetition count, such that more repetition corresponds to fewer unique high-quality tokens. More unique data consistently yields lower validation loss in both domains. However, Math is relatively robust to repetition, whereas Wiki degrades sharply when the repetition count increases beyond 2.

training tokens from a dataset is fixed and repeated tokens are compared with the same amount of unique data. They find that repeated data performs similarly to unique data for up to roughly four repetitions, but do not study how this trade-off varies across domains.

We therefore analyze different high-quality domains separately. Let α denote the fraction of unique high-quality data and e its repetition count. The total fraction of high-quality tokens is

$$\rho = \alpha e.$$

For fixed ρ, increasing e reduces the unique data fraction. This directly compares repeating a smaller dataset e times with using e times as much unique data once under the same token budget.

Figure 6 shows the results for Math and Wiki. The effect of repetition is strongly domain dependent. For Math, increasing the repetition count from 1 to 4 causes only a small increase in validation loss, suggesting that repeated data can largely substitute for additional unique data, consistent with (Muennighoff et al., 2023). In contrast, Wiki degrades clearly with repetition: two repetitions incur only a modest loss increase, while heavier repetition leads to substantially worse performance.

Thus, the effectiveness of repeated data under a fixed token budget depends on the domain rather than following a universal repetition threshold, and the findings are consistent with our observations in Section 3.1. Appendix D shows results for Code and Medical, where validation loss also increases as unique data is replaced by repeated data, with different degrees of degradation across domains.

We also study whether repetition within a high-quality domain affects pretraining performance beyond that domain. We find that replacing unique high-quality tokens with repeated tokens has only a limited effect when the total high-quality and web-data fractions are fixed. The results of evaluating repetition on OOD validation sets are in Appendix D.

4.2 Ablating Various Learning-Rate Schedules

Finally, we examine whether the optimal repetition count depends on the learning-rate schedule. We compare a warmup-stable-decay (WSD) schedule with the decay phase starting at 0%, 25%, 50%, or 75% of training, as well as a constant learning rate. For each schedule, we vary the fraction of unique high-quality data and the repetition count while keeping all other training settings unchanged.

As shown in Figure 7, the learning-rate schedule has a clear effect on the optimal repetition count. Earlier decay causes the validation loss to increase after fewer repetitions, whereas delaying the decay shifts this increase to a larger repetition count. With a constant learning rate, the model tolerates the largest number of repetitions before degradation occurs.

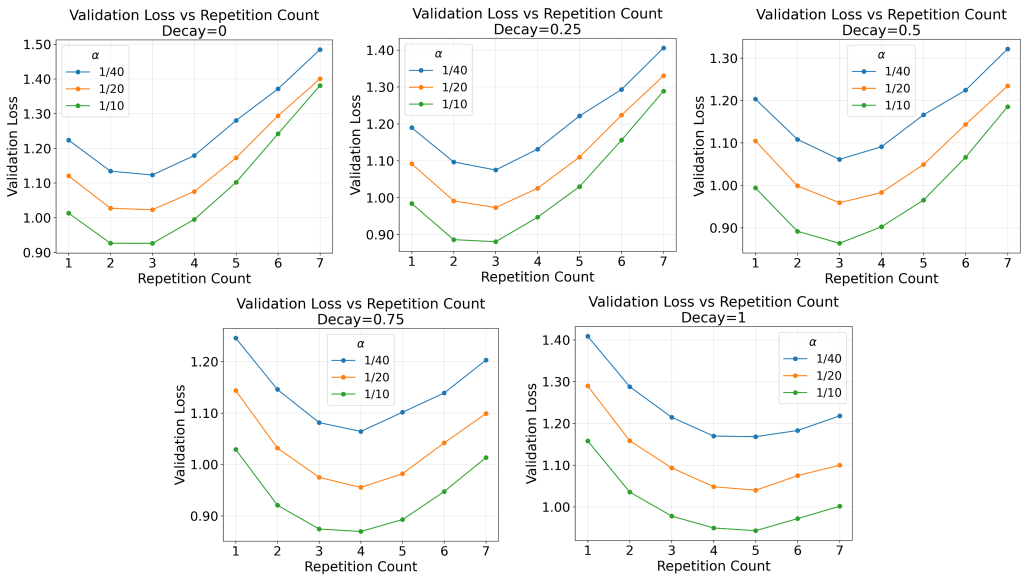


Figure 7: **Effect of the learning-rate schedule on the optimal repetition count.** From left to right: WSD with decay starting at 0%, 25%, 50%, and 75% of training, and a constant learning rate. Later or no decay supports more repetitions before degradation.

A possible explanation is that repeated samples seen during the low-learning-rate stage can be fitted more closely, increasing the tendency to memorize sample-specific patterns. Delaying the decay keeps training at a relatively high learning rate for longer, which may reduce this effect through stronger optimization noise and implicit regularization. Overall, although excessive repetition eventually degrades performance across all schedules, the learning-rate schedule determines how much repetition the model can tolerate before this degradation begins.

5 Conclusion and Future Directions

In this work, we study domain data repetition across model scales while keeping the tokens-per-parameter ratio fixed. We find that the optimal repetition count is strongly negatively correlated with the minimum validation loss of the repeated domain: domains with lower validation loss generally support more repetitions. At a fixed tokens-per-parameter ratio, the optimal repetition count increases mildly with model size, while remaining nearly insensitive to the fraction of unique high-quality tokens over the range considered. These observations suggest that repetition counts selected on smaller proxy models with the same TPP can provide conservative estimates for larger models. Our theoretical analysis further provides an explanation for the observed dependence on validation loss.

Our current setup repeats only one high-quality dataset in each training run. Future work should consider mixtures in which multiple domains are repeated simultaneously and study their interactions. Another direction is to develop a quantitative scaling rule that predicts the optimal repetition count of a large model from small-model results, enabling more accurate and efficient data-recipe selection.

AI use statement

In this work, we used generative AI tools to improve the writing, including polishing the language and correcting grammatical errors. Additionally, we used generative AI tools to assist with the visualization of the experimental framework in Figure 2. We did not use generative AI tools to conduct experiments or generate experimental results. All AI-assisted text was reviewed and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, and results.

REFERENCES

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

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*, 2022.

François Charton and Julia Kempe. Emergent properties with repeated examples. *arXiv preprint arXiv:2410.07041*, 2024.

Jessica Chudnovsky, Joshua Kazdan, Noam Levi, Rylan Schaeffer, Yegor Denisov-Blanch, Bo He, Mehmet Donmez, Sanmi Koyejo, and David Donoho. Internal data repetition destroys language models. *arXiv preprint arXiv:2606.24998*, 2026.

Rui Dai and Shuran Zheng. Explaining data mixing scaling laws. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=joReaAnwnH.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, et al. Climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. *arXiv preprint arXiv:2504.13161*, 2025.

Sumanth Doddapaneni, Gowtham Ramesh, Mitesh Khapra, Anoop Kunchukuttan, and Pratyush Kumar. A primer on pretrained multilingual language models. *ACM Computing Surveys*, 57(9): 1–39, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation. In *International Conference on Machine Learning*, pp. 12895–12915. PMLR, 2024.

Xinran Gu, Kaifeng Lyu, Jiazheng Li, and Jingzhao Zhang. Data mixing can induce phase transitions in knowledge acquisition. *arXiv preprint arXiv:2505.18091*, 2025.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. Wiki-40b: Multilingual language model dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 2440–2452, 2020.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 30016–30030, 2022.

Dayal Singh Kalra and Maissam Barkeshli. Quantifying hyperparameter transfer and the importance of embedding layer learning rate. *arXiv preprint arXiv:2605.21486*, 2026. doi: 10.48550/arXiv.2605.21486.

Feiyang Kang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia. Autoscale: Scale-aware data mixing for pre-training llms. *arXiv preprint arXiv:2407.20177*, 2024.

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.

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.

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 (Volume 1: Long Papers)*, pp. 8424–8445, 2022.

Binghui Li, Fengling Chen, Zixun Huang, Lean Wang, and Lei Wu. Functional scaling laws in kernel regression: Loss dynamics and learning rate schedules. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=dpllevHMbc.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

R Li, LB Allal, Y Zi, N Muennighoff, D Kocetkov, C Mou, M Marone, C Akiki, J Li, J Chim, et al. Starcoder: May the source be with you! *Transactions on machine learning research*, 2023.

Licong Lin, Jingfeng Wu, Sham M. Kakade, Peter Bartlett, and Jason D. Lee. Scaling laws in linear regression: Compute, parameters, and data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=PH7sdEanXP.

Fengze Liu, Weidong Zhou, Binbin Liu, Ping Guo, Zijun Wang, Bingni Zhang, Yifan Zhang, Yifeng Yu, Xiaohuan Zhou, and Taifeng Wang. Infolaw: Information scaling laws for large language models with quality-weighted mixture data and repetition. *arXiv preprint arXiv:2605.02364*, 2026.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. In *The Thirteenth International Conference on Learning Representations*, 2024.

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.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text. In *International Conference on Learning Representations*, volume 2024, pp. 20357–20379, 2024.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Anastasiia Sedova, Skyler Seto, Natalie Schluter, and Pierre Ablin. Scaling laws for mixture pre-training under data constraints. *arXiv preprint arXiv:2605.12715*, 2026.

Mustafa Shukor, Louis Bethune, Dan Busbridge, David Grangier, Enrico Fini, Alaaeldin El-Nouby, and Pierre Ablin. Scaling laws for optimal data mixtures. *arXiv preprint arXiv:2507.09404*, 2025.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Boqian Wu, Qiao Xiao, Patrik Okanovic, Tomasz Sternal, Maurice van Keulen, Mykola Pechenizkiy, Elena Mocanu, Torsten Hoefler, and Decebal Constantin Mocanu. When data is scarce: Scaling sparse language models with repeated training. *arXiv preprint arXiv:2606.01155*, 2026.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. To repeat or not to repeat: Insights from scaling llm under token-crisis. *Advances in Neural Information Processing Systems*, 36:59304–59322, 2023.

Tingkai Yan, Haodong Wen, Binghui Li, Kairong Luo, Wenguang Chen, and Kaifeng Lyu. Larger datasets can be repeated more: A theoretical analysis of multi-epoch scaling in linear regression. *arXiv preprint arXiv:2511.13421*, 2025.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Related Work

Data Repetition in Single-Dataset Training. Early studies mainly examine multi-epoch training on a single dataset. One line of work extends the Chinchilla scaling law (Hoffmann et al., 2022) and develops scaling laws for repeated data. Muennighoff et al. (2023) model the validation loss under repetition using exponential decay and show that a few epochs of repetition can perform similarly to fresh data, while heavier reuse yields diminishing returns. Lovelace et al. (2026) further extend this formulation by modeling the subsequent increase in validation loss with an additive overfitting penalty, achieving more accurate predictions. Another line of work studies the factors that determine repetition-induced degradation. Xue et al. (2023) identify dataset size, model size, and training objective as key factors, while Charton & Kempe (2024) show that repetition can improve generalization on some synthetic mathematical tasks. Yan et al. (2025) theoretically prove that larger datasets can generally support more reuse. Beyond exact duplication, Kazdan et al. (2026) find that semantic duplication can also cause overfitting, and Wu et al. (2026) show that sparse models benefit from more repeated epochs than dense models. However, these studies mainly repeat an entire dataset, whereas practical LLM pretraining often repeats only selected domains within data mixtures.

Data Repetition in Pretraining Mixtures. More recent work studies repetition within pretraining mixtures, where a limited subset is reused while the remaining data remain unique. Hernandez et al. (2022) study a setting in which 10% of the training tokens are drawn from a repeatedly reused subset and the remaining 90% are unique. They find that repetition can cause non-monotonic degradation, with intermediate repetition levels being particularly harmful. Extending this analysis, Chudnovsky et al. (2026) jointly vary the subset size and repetition count, showing that a moderately sized subset repeated a moderate number of times can be more harmful than either a larger subset repeated fewer times or a smaller subset repeated more times. Sedova et al. (2026) study mixture pretraining with limited target-domain data and abundant generic data. They show that generic data mitigates overfitting from repeated target-domain data and that the optimal repetition count depends on the amount of target data, compute budget, and model scale. However, existing cross-scale studies do not preserve the tokens-per-parameter ratio. We instead study repetition under a fixed $\mathrm{TPP} = D/N$, such that the training-token budget grows proportionally with model size.

Data Mixture Optimization for LLMs. Data mixture optimization studies how to allocate a fixed training budget across domains by modeling their relative value and interactions. Existing methods estimate domain weights using proxy training, predictive models, or scaling laws, and transfer the resulting mixture to larger models (Xie et al., 2023; Fan et al., 2024; Liu et al., 2024; Diao et al., 2025; Ye et al., 2025; Kang et al., 2024; Shukor et al., 2025). These methods generally assume sufficient unique data in each domain and therefore do not model repetition. Recent works by Liu et al. (2026) and Sedova et al. (2026) relax this assumption by deriving repetition-aware mixture scaling laws that jointly characterize domain allocation and data reuse.

B General Implementation Details

In this section, we describe the datasets and training hyperparameters used in our experiments.

B.1 Details of Datasets

Our pretraining corpus consists of general web data and several high-quality domain-specific datasets. The general web data contains content from diverse sources, covering a broad range of topics and writing styles. The high-quality data includes four domains:

- **Code data.** A multilingual programming corpus covering a range of programming languages and software-development scenarios.
- **Math data.** A bilingual mathematical corpus containing educational materials, mathematical problems, and step-by-step solutions across different difficulty levels.
- **Wiki data.** Encyclopedic text covering diverse concepts, entities, and factual knowledge.
- **Medical data.** A collection of medical and health-related content covering biomedical knowledge, clinical topics, and health education.

B.2 Details of Training Hyperparameters

The total training-token budget was scaled proportionally with model size under a fixed tokens-per-parameter ratio. Following Bi et al. (2024), we set the learning rate and global batch size for different model sizes according to power-law functions of the model size. The learning rate was linearly warmed up for the first 200 optimization steps and was then held constant for the remainder of training. We used an attention dropout rate of 0.1. Optimization used the Muon optimizer (Liu et al., 2025).

C Additional Results for Section 3.1

In this section, we present additional results of Section 3.1.

Figures 8, 9, 10, and 11 present the complete results for Math, Wiki, Code, and Medical. Among the four domains, Math supports the most repetition, with an optimal repetition count of about 5–6, followed by Code at about 4–5 repetitions, while Wiki and Medical typically reach their minima at about 3–4 repetitions. For each domain, the optimal repetition count increases with model size. In contrast, for a fixed model size, the optimal repetition count remains nearly unchanged across different values of α. Overall, all experimental results are consistent with the observations in the main text.

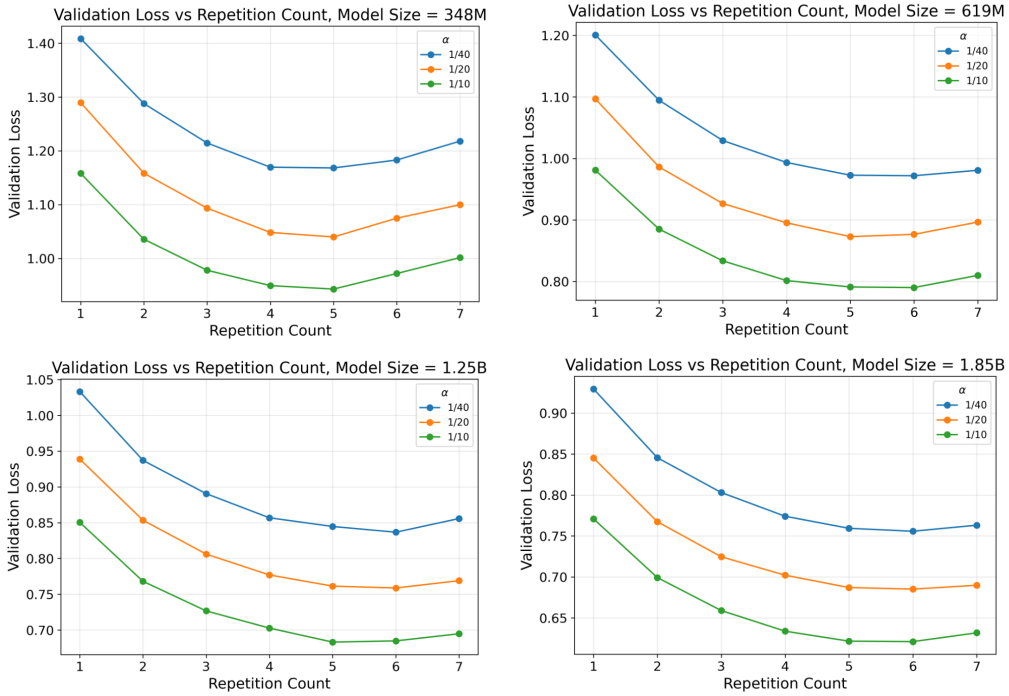


Figure 8: **Final validation loss across repetition counts for Math.** Each panel corresponds to a different model size, and each curve represents a different fraction of unique high-quality data. All configurations are trained for the full token budget associated with the corresponding model size. The validation loss is generally minimized between 5 and 6 repetitions. Increasing the unique data fraction reduces the absolute validation loss but has little effect on the optimal repetition count, which increases only mildly with model size.

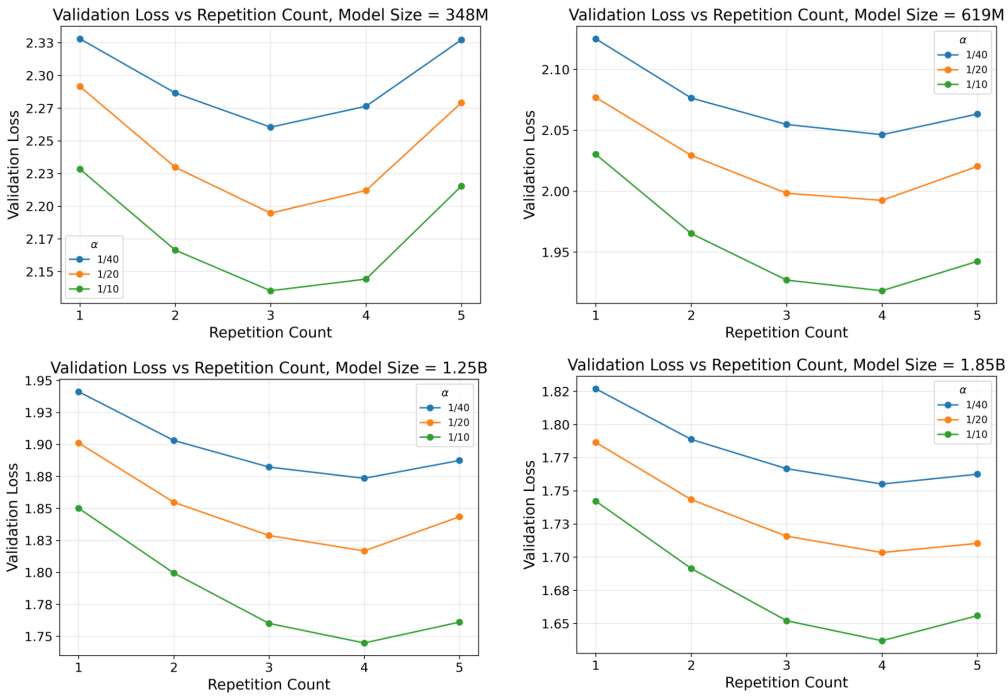


Figure 9: **Final validation loss across repetition counts for Wiki.** Compared with Math, Wiki reaches its minimum validation loss earlier, typically after 3 to 4 repetitions, while showing the same stability across unique data fractions and mild increase with model size.

D Additional Results for Section 4.1

In this section, we present additional results of Section 4.1.

Figure 12 shows the results under the same fixed-token-budget setup as in the main text. For each curve, the total fraction of high-quality training tokens is fixed, while increasing the repetition count reduces the amount of unique high-quality data. This directly compares repeated data with an equivalent number of fresh unique tokens.

Code and Medical behave similarly to Wikipedia, with validation loss increasing more noticeably as more unique data is replaced by repeated data. These results further show that the extent to which repeated data can match fresh data under a fixed token budget depends strongly on the data domain.

We also study whether repetition within a high-quality domain affects pretraining performance beyond that domain. To isolate the effect of repetition, it is important to control the amount of web data used during training. If we instead fix the amount of unique high-quality data and increase its repetition count, the total amount of high-quality data increases accordingly, leaving fewer training tokens for web data. Any change in OOD performance would then confound the effect of repetition with the change in web data.

We therefore adopt the fixed-domain-fraction setup from Section 4.1. Specifically, we fix the total fraction of Math tokens, and hence also the amount of web data, while varying the repetition count by changing the amount of unique high-quality data. This allows us to isolate how replacing unique high-quality tokens with repeated tokens affects performance outside the repeated domain.

We evaluate the trained models on two OOD validation sets, ArXiv and News. As shown in Figure 13, OOD validation loss remains largely stable across repetition counts, with only small differences between 1, 2, and 4 repetitions across model sizes and high-quality data fractions.

These results suggest that, when the total fractions of high-quality and web tokens are fixed, replacing unique high-quality tokens with repeated tokens mainly affects performance within the repeated domain, while having limited effect on OOD pretraining performance.

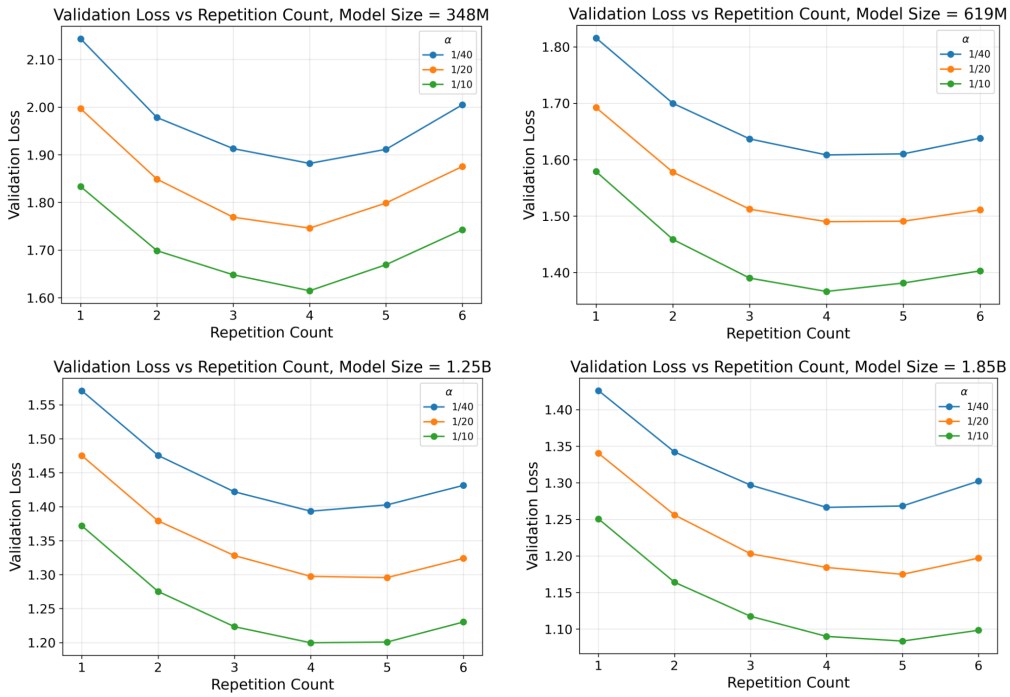


Figure 10: **Final validation loss across repetition counts for Code.** Code generally reaches its minimum validation loss after 4 to 5 repetitions. The optimal repetition count remains stable across unique data fractions and increases mildly with model size.

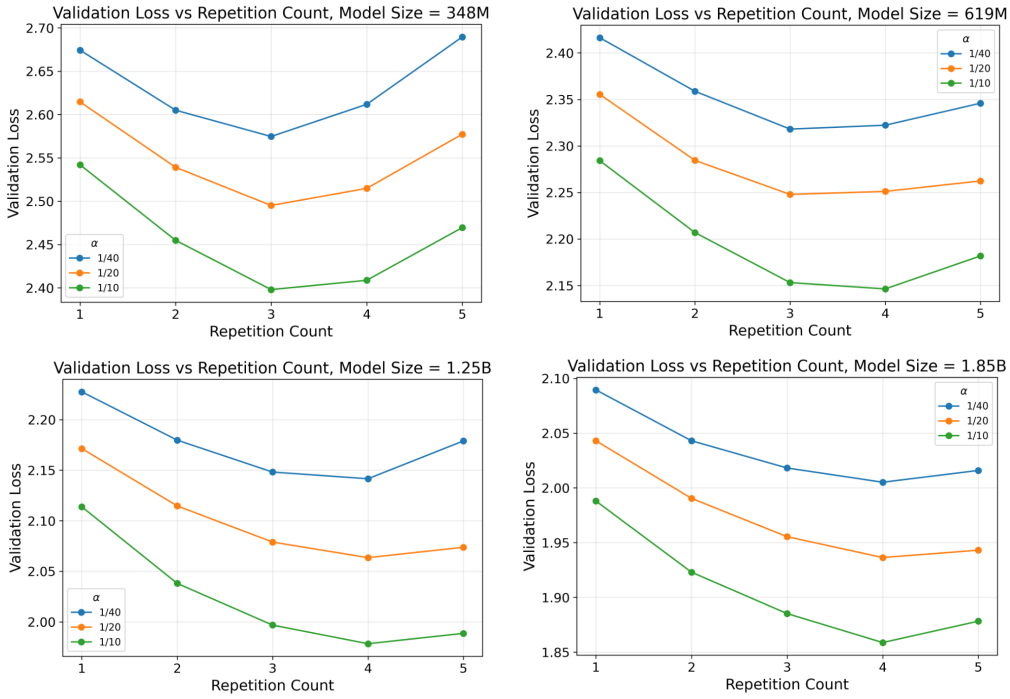


Figure 11: **Final validation loss across repetition counts for Medical.** Medical generally reaches its minimum validation loss after 3 to 4 repetitions. As in the other domains, the optimal repetition count is largely insensitive to the unique data fraction and increases only mildly with model size.

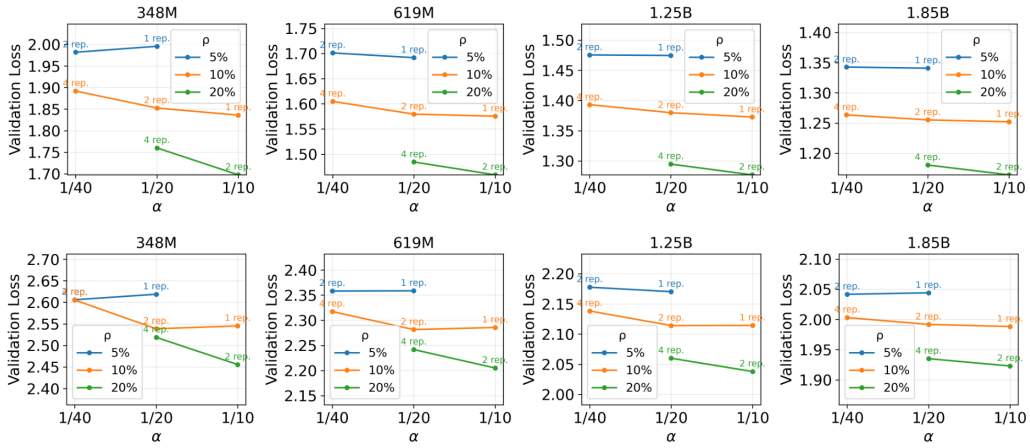


Figure 12: **Additional results at fixed total high-quality data fractions.** Top: Code. Bottom: Medical. Code and Medical behave similarly to Wikipedia and show a clearer degradation as more unique data is replaced by repeated data.

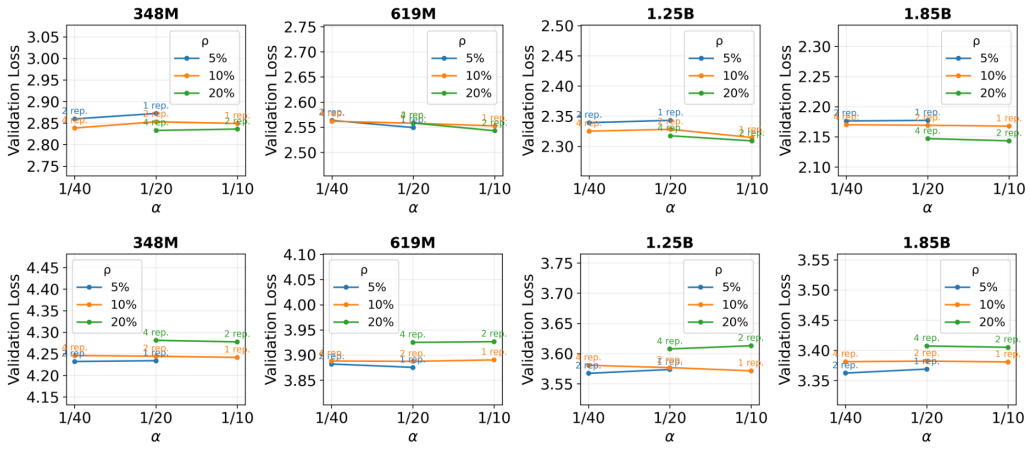


Figure 13: **OOD validation loss at fixed total high-quality data fractions.** Top: ArXiv. Bottom: News. Each line fixes the total fraction of high-quality tokens, while the annotations indicate the corresponding repetition counts. Unlike the in-domain validation loss, the OOD loss changes only slightly as unique tokens are replaced with repeated tokens and shows no consistent monotonic relationship with the repetition count.

E PROOF OF THEOREM 3.2

Proof of Theorem 3.2. We write $U = U_N$ and denote by $\mathcal{I}$ the fixed set of retained high-quality coordinates. We first derive the coordinate-wise SGD dynamics. Fix $i \in \mathcal{I}$, and let

$$M_i = \sum_{\ell=1}^{U} \mathbf{1}_{\{\mathbf{x}_\ell^1 = \mathbf{e}_i\}} \sim \mathrm{Binomial}(U, p_i).$$

Conditional on $M_i = m \geq 1$, define

$$\bar{\varepsilon}_i = \frac{1}{m} \sum_{\ell:\, \mathbf{x}_\ell^1 = \mathbf{e}_i} \varepsilon_\ell^1, \qquad S_i^2 = \frac{1}{m} \sum_{\ell:\, \mathbf{x}_\ell^1 = \mathbf{e}_i} (\varepsilon_\ell^1 - \bar{\varepsilon}_i)^2.$$

For simplicity, set $z_{t,i} = \theta_{t,i} - \theta_{1,i}^* - \bar{\varepsilon}_i$. If the sample drawn at step t has feature $\mathbf{e}_i$, then

$$z_{t+1,i} = (1-\eta) z_{t,i} + \eta(\varepsilon_{J_t}^1 - \bar{\varepsilon}_i),$$

and otherwise $z_{t+1,i} = z_{t,i}$. Taking expectation over $J_t \sim \mathrm{Unif}\{1, \ldots, U\}$, we obtain

$$\mathbb{E}[z_{t+1,i} \mid z_{t,i}, \mathcal{D}_1] = a_m z_{t,i}, \qquad a_m = 1 - \frac{\eta m}{U}, \tag{1}$$

$$\mathbb{E}[z_{t+1,i}^2 \mid z_{t,i}, \mathcal{D}_1] = b_m z_{t,i}^2 + \frac{\eta^2 m}{U} S_i^2, \qquad b_m = 1 - \frac{\eta(2-\eta)m}{U}. \tag{2}$$

Starting from $z_{0,i} = -(\theta_{1,i}^* + \bar{\varepsilon}_i)$ and iterating Equations 1 and 2 for eU steps gives

$$\begin{aligned}
\mathbb{E}[z_{eU,i} \mid \mathcal{D}_1] &= -a_m^{eU}(\theta_{1,i}^* + \bar{\varepsilon}_i), \\
\mathbb{E}[z_{eU,i}^2 \mid \mathcal{D}_1] &= b_m^{eU}(\theta_{1,i}^* + \bar{\varepsilon}_i)^2 + \frac{\eta}{2-\eta} S_i^2 (1 - b_m^{eU}).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}[(\theta_{eU,i} - \theta_{1,i}^*)^2 \mid \mathcal{D}_1] = b_m^{eU}(\theta_{1,i}^* + \bar{\varepsilon}_i)^2 - 2a_m^{eU}\bar{\varepsilon}_i(\theta_{1,i}^* + \bar{\varepsilon}_i) \\
+ \bar{\varepsilon}_i^2 + \frac{\eta}{2-\eta} S_i^2 (1 - b_m^{eU}).
\end{aligned} \tag{3}$$

If $m = 0$, or $i \notin \mathcal{I}$, the corresponding coordinate remains at zero.

We next compute the expected population risk. For $m \geq 1$,

$$\mathbb{E}[\bar{\varepsilon}_i^2 \mid M_i = m] = \frac{\sigma^2}{m}, \qquad \mathbb{E}[S_i^2 \mid M_i = m] = \sigma^2 \left(1 - \frac{1}{m}\right).$$

Combining these identities with $p_i \mathbb{E}[(\theta_{1,i}^*)^2] = i^{-\beta}$ in Equation 3, we have

$$2\overline{\mathcal{R}}_N(e; \sigma^2) = B_N(e) + \sigma^2 V_N(e), \tag{4}$$

where, with $b_0 = 1$,

$$B_N(e) = \sum_{i \notin \mathcal{I}} i^{-\beta} + \sum_{i \in \mathcal{I}} i^{-\beta} \mathbb{E}[b_{M_i}^{eU}], \tag{5}$$

$$V_N(e) = 1 + \sum_{i \in \mathcal{I}} p_i \mathbb{E}[v_e(M_i) \mathbf{1}_{\{M_i \geq 1\}}], \tag{6}$$

$$v_e(m) = \frac{1 - 2a_m^{eU} + b_m^{eU}}{m} + \frac{\eta}{2-\eta}\left(1 - \frac{1}{m}\right)(1 - b_m^{eU}). \tag{7}$$

Equations 5–7 also define the continuous extension to real $e \geq 1$.

We now identify the leading terms of the risk derivative. Denote

$$\pi_{i,m} = \mathbb{P}(M_i = m), \qquad a_1 = 1 - \frac{\eta}{U}, \qquad b_1 = 1 - \frac{\eta(2-\eta)}{U}.$$

Note that $\pi_{i,1} = Up_i(1-p_i)^{U-1} > 0$. Since $0 < b_m < a_m < 1$ and both decrease strictly with m, the $m = 1$ term decays slowest. Differentiating Equation 5 therefore gives

$$-B'_N(e) = C_B b_1^{eU}\{1 + o_e(1)\}, \qquad C_B = -U\log(b_1)\sum_{i\in\mathcal{I}} i^{-\beta}\pi_{i,1} > 0. \tag{8}$$

Likewise, differentiating Equation 7 gives

$$V'_N(e) = U\sum_{i\in\mathcal{I}} p_i \sum_{m=1}^{U} \pi_{i,m}\Bigg[-\frac{2\log(a_m)}{m} a_m^{eU} + \log(b_m)\left\{\frac{1}{m} - \frac{\eta}{2-\eta}\left(1-\frac{1}{m}\right)\right\} b_m^{eU}\Bigg].$$

Here the slowest term is a_1^{eU}, and hence

$$V'_N(e) = C_V a_1^{eU}\{1 + o_e(1)\}, \qquad C_V = -2U\log(a_1)\sum_{i\in\mathcal{I}} p_i\pi_{i,1} > 0. \tag{9}$$

It remains to optimize Equation 4. The bias B_N is strictly decreasing, whereas V_N is bounded. It follows that $e_N^*(\sigma^2) \to \infty$ as $\sigma^2 \downarrow 0$. Moreover, because $a_1 > b_1$, Equation 9 eventually dominates Equation 8 for every fixed $\sigma^2 > 0$. On the other hand, the risk derivative at $e = 1$ is negative for sufficiently small σ^2. Thus the minimizer is finite and interior in the small-noise regime. Its first-order condition is

$$0 = -C_B b_1^{e_N^* U}\{1 + o_{\sigma^2}(1)\} + \sigma^2 C_V a_1^{e_N^* U}\{1 + o_{\sigma^2}(1)\}.$$

Therefore,

$$\left(\frac{b_1}{a_1}\right)^{e_N^* U} = \sigma^2 \frac{C_V}{C_B}\{1 + o_{\sigma^2}(1)\},$$

which implies

$$e_N^*(\sigma^2) = \frac{\log(1/\sigma^2) + \log(C_B/C_V)}{U\log(a_1/b_1)} + o_{\sigma^2}(1). \tag{10}$$

Theorem 3.2 follows immediately.

We finally verify monotonicity. Let $F(e,\sigma^2) = B'_N(e) + \sigma^2 V'_N(e)$. Differentiating the finite exponential sums once more gives

$$\frac{\mathrm{d}}{\mathrm{d}e}\log\left(\frac{-B'_N(e)}{V'_N(e)}\right) = U\log(b_1/a_1) + o_e(1) < 0.$$

Thus the large-e stationary point is unique. At the root in Equation 10,

$$\partial_e F(e_N^*, \sigma^2) = C_B b_1^{e_N^* U}\, U\log(a_1/b_1)\{1 + o_{\sigma^2}(1)\} > 0.$$

The implicit-function theorem yields

$$\frac{\mathrm{d}e_N^*}{\mathrm{d}\sigma^2} = -\frac{V'_N(e_N^*)}{\partial_e F(e_N^*, \sigma^2)} < 0$$

for all sufficiently small σ^2. This completes the proof. □