REWRITING SCALING LAWS FOR SYNTHETIC DATA WITH COMPUTE AND DATA CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic rewrites, where a language model rewrites a natural document into a target style, are now a standard ingredient of pretraining corpora. Yet critical aspects of their use remain poorly understood, such as how much of the training data should be rewritten, given that generating rewrites costs compute that could instead train the model, and whether a model trained on rewrites can outperform the rewriter. We study these two questions by proposing a rewriting scaling law that predicts loss from model size, training tokens, the fraction of training tokens that are rewrites, and the number of epochs over the natural data, charging the compute spent generating rewrites. We fit it in two settings. In the compute-limited setting, natural data is plentiful and rewrites come from a third-party rewriter trained with more compute than our largest run. We find that the estimated optimal rewrite fraction increases with compute, from under 10% at 3×10^{18} FLOPs to 32% at 10^{22} FLOPs. Our scaling forecast extrapolates $33\times$ in FLOPs to train a 22B-parameter model, whose final loss is within 1.1% of our prediction. This model achieves a better BPB than its rewriter on several web domains at one-tenth the rewriter's training compute. In the data-limited setting, natural data is a fixed seed set and the rewriter is a model of the same size as the target, pretrained on that seed set. Again the optimal rewrite fraction increases with the compute budget. We also demonstrate that a model trained on rewrites of its own pretraining data, generated by a rewriter of the same size, beats the best run on natural data alone at every model size up to 3B. Our results show that synthetic rewriting can be studied through the scaling law lens: that optimal methodological choices are predictable and scaling does not necessarily break down beyond the rewriter's capabilities.

1 INTRODUCTION

Large language models have a host of remarkable capabilities, many of which improve smoothly during pretraining and grow as model parameters and training tokens increase, in a manner that is predictable via scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). Early language models were pretrained using corpora of text that were primarily comprised of *natural data*, text generated either manually by a human or generated programmatically. Common sources of such tokens included web pages collected from internet crawls, GitHub repositories, and academic publications (Gao et al., 2020). However, the amount of compute available for training has grown much faster than the amount of high quality, natural pretraining data (Villalobos et al., 2022; Kim et al., 2025).

These constraints on the availability of large-scale natural corpora of text have motivated using *synthetic rewrites*, documents generated by prompting a language model with a natural document and an instruction to write a different version of the document. Common instructions for synthetic rewrites include rewriting in the manner of a Wikipedia article (Maini et al., 2024), converting math web pages into synthetic math textbook-style exposition (Mahabad et al., 2025), synthesizing or summarizing lengthy source material (Maini et al., 2025), and transforming a lower quality source into a higher quality output using the model's knowledge (Su et al., 2024; Nguyen et al., 2025).

Synthetic rewrites of natural documents have become a standard ingredient in many modern pretraining corpora (Maini et al., 2025; Su et al., 2024; Nguyen et al., 2025; Niklaus et al., 2026). However, their use does have several drawbacks. First, synthetic rewrites are costly to generate: for a given model size, rewriting a given document involves a significant fraction of the compute

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.

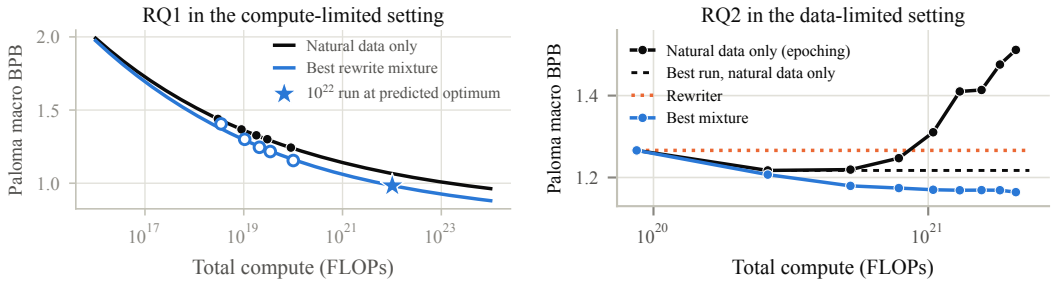

Figure 1: *(Left; compute-limited) Effective rewriting recipes are predictable via scaling laws:* best possible Paloma BPB (lower is better; allowing third-party rewrites) vs FLOPs (accounting for rewrite compute costs) as compared to using natural data only. Unfilled blue points are imputed via linear interpolations of the loss of the closest rewrite fractions that we evaluate, whereas solid points represent observed training runs. *(Right; data-limited) The trained model surpasses its rewriter:* final Paloma BPB vs FLOP for a 3B target model trained using only natural data or synthetic rewrites coming from a fixed 5B-token natural data corpus. Using natural data only, which necessitates epoching, results in worse loss after 3 epochs. The best proportion of rewrites from a bootstrapped rewriter improves over epoching on the natural seed set alone at every budget.

of training on that document. Second, synthetic rewrites are necessarily generated by a model with a given amount of complexity and capability, and it is unclear the extent to which that bounds the capabilities of a model trained on these synthetic rewrites.

In this paper we develop a *rewriting scaling law* that takes as input the *rewrite fraction*, the fraction of training tokens that are synthetic rewrites, and predicts the expected loss of a model trained at that rewrite fraction. When extrapolating this scaling law to predict the loss-minimizing fraction at a given compute budget, we account for the *cost-adjusted training FLOPs*, the amount of FLOPs available for training after accounting for the cost of generating synthetic rewrites.

With this scaling law we investigate two specific research questions:

RQ1 *Does our scaling law accurately predict the optimal rewrite fraction and its achieved loss?*
RQ2 *Should we include rewrites when targeting capabilities that surpass those of the rewriter?*

We study these questions in distinct settings characteristic of language model training paradigms in which the primary constraint is either compute (Hoffmann et al., 2022) or data (Kim et al., 2025).

In the first setting, which we term *compute-limited*, we focus on a scenario where natural data is unbounded. We assume an *external rewriter*, a rewriting model that is trained with more total compute than our largest training run. This setting exemplifies the small-scale scientific workflow.

In the second setting, which we term the *data-limited* setting, we focus on a scenario in which natural data is bounded. We also assume that we only have access to a *bootstrapped rewriter* which is trained on a subset of the available data and for fewer FLOPs than a given candidate training run. This setting exemplifies the frontier-scale workflow in which the objective is to train a model more capable than any previous model.

We find the following results:

- *Our scaling law accurately predicts the optimal rewrite fraction and achieved loss at large budgets.* In the compute-limited setting, we fit our rewriting scaling law at low budgets and extrapolate to a set of 10^{22}-FLOP runs. We find that the predicted optimal rewrite fraction results in the lowest held-out loss among tested models at the 10^{22}-FLOP scale, achieving loss within 1.1% of the predicted loss. In the data-limited setting, a scaling law fit to low budgets predicts a rewrite fraction at large held-out budgets whose loss is within 1.14% of that of the best observed rewrite fraction. In both settings, the optimal rewrite fractions increase with scale and are task-specific, differing between math, code, question-answering, and general web text.

- *Models can achieve and exceed their rewriter model's performance.* In the compute-limited setting, pretraining a model for 10^{22} FLOPs—where 32% of its training data is rewritten—outperforms the external rewriter (SmolLM2-1.7B-Instruct) on 10 out of 16 general web text sources (using roughly one-tenth of the rewriter's training compute). In the data-limited setting, where the bootstrapped rewriter is always trained with fewer FLOPs than the target model, models trained on a mixture containing rewrites consistently outperform the rewriter.

Our results give evidence that synthetic rewriting can be studied through the scaling law lens, showing both that optimal methodological choices are predictable and that scaling does not necessarily break down beyond the capabilities of the rewriter.

2 METHODOLOGY

In this section we first present the rewriting scaling law we use to predict the expected loss of a model trained with a given amount of synthetic rewrites in the training set. We then describe how we practically fit and use this scaling law to predict optimal rewrite fractions, describing how we account for the cost of compute to generate the synthetic rewrites.

2.1 A REWRITING SCALING LAW

In a given experiment in both the compute- and data-limited settings, we pretrain an N-parameter *target model* on a *corpus* with D total tokens composed of a combination of *natural tokens* and *synthetic rewrites* thereof. The set of natural tokens in a given corpus is drawn from a *seed set* of T natural tokens, potentially with replacement (epoching). The synthetic rewrite tokens are rewrites of the natural tokens selected for the corpus, without epoching. We define the *rewrite fraction* $R \in [0, 1]$ for a given corpus as the fraction of tokens in the corpus that are synthetic rewrites.

Our *rewriting scaling law* predicts that an N-parameter target model trained on a D-token corpus with seed set size T and rewrite fraction R has held-out loss L as follows:

$$L(N, D, R, T) = \underbrace{\frac{A}{N^\alpha} + \frac{B}{D^\beta}}_{\text{Chinchilla terms}} + \underbrace{E(R)}_{\substack{\text{rewrite-fraction-dependent} \\ \text{irreducible loss}}} + \underbrace{F(R, D, T)}_{\text{repetition penalty}}. \tag{1}$$

The terms $A/N^\alpha + B/D^\beta$ are the same as in the classic Chinchilla-style scaling law of Hoffmann et al. (2022), which model the loss contributions of using a certain parameter count and dataset size.

The term $E(R)$ represents the irreducible loss as a function of the rewrite fraction R. Appendix L defines this as an approximation of the KL divergence between the training and evaluation distributions. The specific functional form for this *rewrite-fraction-dependent irreducible loss* is given as $E(R) = E_0 - \lambda_n \log(1 - (1 - \rho_n)R) - \lambda_s \log(\rho_s + (1 - \rho_s)R)$, where λ_n, λ_s, ρ_n, ρ_s, and E_0 are all fitted parameters. Additional details can be found in Appendix L.

The term $F(R, D, T)$ represents the cost of epoching natural data multiple times, and is drawn from Equation (7) in Lovelace et al. (2026). This *repetition penalty* term is defined as $F(R, D, T) = P\,(1-R)^s\,\big((1-R)D/T - 1\big)_+$, where P and s are fitted parameters. The repetition penalty is only nonzero when the natural tokens are epoched, so it appears only in the data-limited setting.

2.2 PREDICTING OPTIMAL REWRITE FRACTIONS AT A COMPUTE BUDGET

Once we fit our rewriting scaling law, our goal is to choose the rewrite fraction that minimizes loss for a fixed compute budget. Since synthetic rewrites are not free, the compute budget must account for both the cost of training the target model and the cost of generating synthetic rewrites.

We approximate the FLOPs for training a target model with N parameters on a corpus of size D with rewrite fraction R using synthetic data generated by a rewriter with N_r parameters with:

$$C_{\text{total}} := \underbrace{6ND}_{\text{target-model training}} + \underbrace{2N_rDR}_{\text{source prefill}} + \underbrace{2N_rDR}_{\text{rewrite decoding}}. \tag{2}$$

This C_{total} approximation makes several assumptions. First, we use the standard assumptions that forward and backward passes are respectively $2N$ and $4N$ FLOPs per token (Kaplan et al., 2020), giving $6ND$ target-model training FLOPs. Second, we assume that the length of a synthetic rewrite is the same as the length of the input document, and that these lengths are large relative to other elements of the prompt; this gives $2N_rDR$ FLOPs for forward passes (prefill) on the natural tokens used for rewriting, and an additional $2N_rDR$ FLOPs for forward passes (decode) when generating the synthetic rewrites.[1] We also assume no amortization of the cost of generating rewrites across multiple different target model training runs.

The *optimal rewrite fraction* R^* minimizes the true loss at a given compute budget. We denote our estimate of it as $\widehat{R}^*$, which is the minimizer of our fitted scaling law:

$$\widehat{R}^* = \arg\min_{R\in[0,1]} \min_{N,\,D} L(N, D, R, T) \quad \text{s.t.} \quad 6ND + 4N_rRD = C_{\text{total}}. \tag{3}$$

3 Experiments

We study our research questions in two different settings.

First in the *compute-limited* setting, we assume a fixed FLOP budget, more natural data in our seed set than our largest target model requires ($T \gg D$), and an *external rewriter* trained for more FLOPs than any member of our scaling ladder on a potentially different seed set of data. This setting is motivated by the small-scale scientific workflow.

Next in the *data-limited* setting, we assume a fixed FLOP budget, a limited seed set ($T \ll D$), and for each target model a corresponding *bootstrapped rewriter* trained for fewer FLOPs than that target model on the same seed set. This setting is motivated by the frontier-scale workflow in which the objective is to train a model more capable than any previous model.

Consistent across both settings we find: **(RQ1)** scaling law based extrapolations well predict the optimal fraction of rewrites and the corresponding achieved loss; and **(RQ2)**, in both the compute-limited and data-limited settings, the scaling law predicts and we empirically confirm that it is optimal to include rewrites (i.e., $\widehat{R}^* > 0$) for target models that surpass capabilities of the rewriter.

3.1 Experimental Setup

Architecture and training. All runs use Qwen3 dense models (Yang et al., 2025a) with the Llama 3 tokenizer (Grattafiori et al., 2024), trained with Adam and Hyperball (Wen et al., 2026) under a linear warm-up and linear-decay schedule. Appendix B gives the width-to-depth rule and Appendix C the hyperparameter tuning and transfer recipe.

Data. In the compute-limited setting, we train on FineWeb-Edu (Penedo et al., 2024) with FinePhrase rewrites (Niklaus et al., 2026), which are FineWeb-Edu rewrites from SmolLM2-1.7B (Ben Allal et al., 2025). In the data-limited setting, we train on a 5B-token seed set from FineWeb-Edu and rewrites of that seed set using a model that we train ourselves, also on the seed set. We retain only documents from web-archive (WARC) files dated before 2023 to reduce LM-generated text in the natural corpus (Appendix A). For mixtures with rewrite fraction $0 < R < 1$ we interleave natural and rewritten documents in proportion to R throughout training (Appendix I).

Rewriting protocol. Following FinePhrase (Niklaus et al., 2026), every rewrite converts one natural document into a math exposition, an FAQ, a table, or a tutorial.

Evaluation. Our evaluations include Paloma (Magnusson et al., 2024) and the set of aggregations in the Olmo 3 base easy evaluation suite (Ettinger et al., 2025): QA (ARC, HellaSwag, and MMLU), Math (Minerva MATH, with GSM8K-Platinum and TheoremQA added), and Code (HumanEval, MBPP, and multilingual MBPP) (Clark et al., 2018; Zellers et al., 2019; Hendrycks et al., 2021a;

[1] For FLOPs to be a reasonable proxy for the cost of generating synthetic rewrites, we must also assume that generation is FLOP-bound and not memory-bandwidth-bound; this often holds with sufficiently large batch sizes and short context lengths (Pope et al., 2023), which is tractable when generating synthetic data.

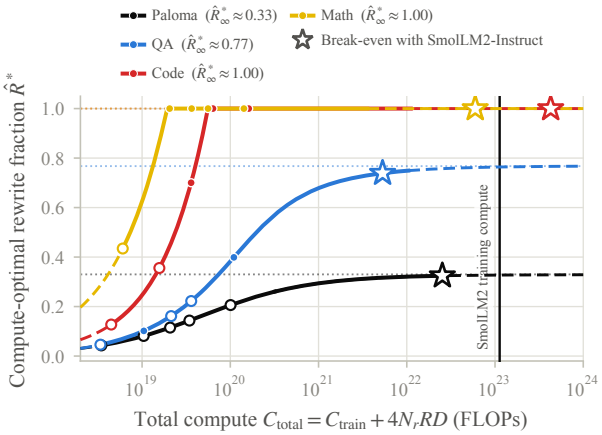

Figure 2: **The estimated compute-optimal rewrite fraction increases with compute on every evaluation.** The curves show $\widehat{R}^*$ across evaluations and FLOP budgets. Solid lines indicate that the compute budget is within the evaluated ladder, while dashed lines are extrapolations. Points mark the fit optimum at each ladder budget; filled points land on a tested rewrite fraction and unfilled points fall between two. Stars mark predicted break-even with SmolLM2-Instruct. The vertical line marks the training FLOPs of SmolLM2-Instruct.

Cobbe et al., 2021; Vendrow et al., 2025; Chen et al., 2023; Hendrycks et al., 2021b; Lewkowycz et al., 2022; Chen et al., 2021; Austin et al., 2021; Cassano et al., 2023). We list the suite membership and aggregation for all evaluations in Appendix D.

On all evaluations, we report bits-per-byte (BPB) at the trained 4096-token context. For Paloma, we report the macro-averaged BPB, BPB averaged within each Paloma domain then averaged across domains (equal-weighting all domains). For the Olmo 3 suites we report gold-answer BPB.

3.2 Compute-Limited Experiments

In the *compute-limited* setting, we assume a fixed FLOP budget, more natural data in our seed set than our largest target model requires ($T \gg D$), and an *external rewriter* trained for more FLOPs than any member of our scaling ladder on a potentially different seed set of data.

Scaling ladder methodology. We train *IsoFLOP ladders*—sets of models with varying size and token counts, each trained at one of a range of fixed training budgets—on FineWeb-Edu at eleven rewrite fractions $R \in \{0, 0.1, \ldots, 1.0\}$. SmolLM2-1.7B (Ben Allal et al., 2025) generates rewrites, which we draw from FinePhrase (Niklaus et al., 2026). We train target models covering training budgets spanning 3×10^{18} to 9×10^{19} FLOPs at all eleven rewrite fractions, and natural-only ($R = 0$) ladders at 1.8×10^{20} and 3×10^{20} FLOPs. We fit our scaling laws with these sets of target models against each of the QA, Math, Code, and Paloma evals. We validate the rewriting scaling law's predictions with validation runs at 10^{21} FLOPs across all eleven rewrite fractions and a set of 10^{22} runs including a run at the predicted $\widehat{R}^*$ optimum (and corresponding N and D optima) for the Paloma eval.

Rewriting methodology. In the compute-limited setting, we use the released FinePhrase SmolLM2-1.7B rewrites. SmolLM2-1.7B was trained for roughly 10^{23} FLOPs.

3.2.1 Compute-Limited Experimental Results

The estimated optimal rewrite fraction increases in FLOP budget and varies between tasks (RQ1). Figure 2 shows the estimated optimal rewrite fraction $\widehat{R}^*$ for each different eval as a function of FLOP budget. For all evals, $\widehat{R}^*$ monotonically increases with FLOP budget. This is consistent with that as the total compute budget (and in particular the size of the target model) grows, the cost of generating synthetic rewrites shrinks relative to the cost of training.

We find that $\widehat{R}^*$ changes across target domains: on Code and Math, $\widehat{R}^*$ rises more quickly than on QA and Paloma. On Code and Math, $\widehat{R}^*$ rises to 1.0 at FLOP budgets within our scaling ladder; on QA and Paloma, our scaling law fit extrapolates asymptotes of 0.77 and 0.32 respectively. SmolLM2-1.7B was trained on FineMath, Stack-Edu, and StarCoderData (Ben Allal et al., 2025; Li et al., 2023), which contain higher quality data for math and code capabilities than in the natural

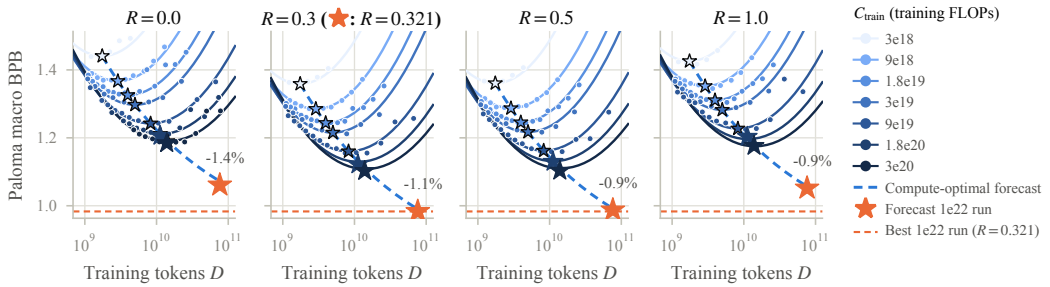

Figure 3: **The surface fit to budgets through 3×10^{20} FLOPs forecasts the 10^{22} runs to within 1.4%.** Each panel is one rewrite fraction R. Dots are runs and curves are the fit IsoFLOP slices, colored by training budget. The $R = 0$ ladders reach 3×10^{20} FLOPs and the $R > 0$ ladders 9×10^{19} FLOPs. Open stars mark the fit optimum at each budget, filled blue stars mark the compute-optimal runs at 1.8×10^{20} and 3×10^{20} FLOPs, and the dashed line extends the compute-optimal forecast to 10^{22} FLOPs. The orange star is the 10^{22} run, held out of the fit, labeled with its measured loss relative to the forecast. The $R = 0.32$ run's forecast error is -1.1% whether the surface is evaluated at $R = 0.3$ or at $R = 0.32$. Figure 14 shows all eleven rewrite fractions.

FineWeb-Edu corpus. The fact that math and code eventually require a rewrite fraction of $R = 1$ suggests that it is easier to learn math and code capabilities by proxy through SmolLM2-1.7B rewrites than from the natural FineWeb-Edu data itself.

The differences in loss between the predicted optimum $\widehat{R}^*$ and suboptimal R points are substantial: at 10^{22} FLOPs, training on rewrites alone ($R = 1$) instead of natural data alone ($R = 0$) reduces Math BPB from 1.33 to 0.80 and Code BPB from 1.04 to 0.53.

The estimated optimal rewrite fraction extrapolated from smaller budgets gives the lowest measured Paloma BPB at 10^{22} FLOPs, across all models we train (RQ1). Our scaling law extrapolation predicts that Paloma BPB at a total budget of 10^{22} FLOPs will be minimized at rewrite fraction $R = 0.32$ (Figure 2). We train four models at 10^{22} FLOPs that share (N, D) and differ in $R \in \{0, 0.32, 0.5, 1.0\}$. Of these, $R = 0.32$ is indeed optimal (Figure 3).

The 10^{22}-FLOP run at the predicted optimum reaches a lower Paloma BPB than its rewriter on 10 of 16 sources (RQ2). SmolLM2-1.7B-Instruct was trained on 11T tokens at $\approx 1.1 \times 10^{23}$ FLOPs. The 10^{22}-FLOP run at $R = 0.32$ reaches a lower Paloma BPB than its rewriter on 10 of the 16 Paloma sources and comes within 0.012 BPB of it on the macro average, 0.983 against 0.972 (Table 1). The sources where the run does not break even with the rewriter are code (which the rewriter's training corpus covers more than our natural corpus FineWeb-Edu (Ben Allal et al., 2025, Figure 2)), social media, and two web mixtures. Table 1 also compares against a 10^{22} run at $R = 0$ which underperforms $R = 0.32$ on most benchmarks, showing that including some rewrites helps surpass the rewriter at this compute budget. The rewriter's BPB therefore does not cap the gains from its rewrites in the compute-limited setting.

3.3 Data-Limited Experiments

In the *data-limited* setting, we assume a fixed FLOP budget, a limited natural seed set ($T \ll D$), and for each target model a corresponding *bootstrapped rewriter* model trained for fewer FLOPs than that target model on the same fixed natural seed set.

Scaling ladder methodology. The seed set is $T = 5$B tokens of FineWeb-Edu. We train target models with 550M, 837M, 1B, and 3B parameters. We measure each target model's total compute budget in *epoch budgets* $W = C_{\text{total}}/6NT$, the compute required to train an N-parameter model, epoching the entire seed set W times. We train models with epoch budgets $W \in \{3, 6, 9, \ldots, 24\}$.

Paloma source	$R = 0$	$R = 0.32$	Rewriter
M2D2 Wikipedia	0.726	**0.716**	0.758
Penn Treebank	0.851	**0.843**	0.882
WikiText-103	0.682	**0.673**	0.711
C4 English	**0.827**	0.828	0.844
Falcon RefinedWeb	0.894	**0.887**	0.897
Gab	1.719	**1.579**	1.587
C4 100 Domains	**0.790**	0.793	0.797
M2D2 S2ORC	0.766	**0.760**	0.764
mC4	0.841	**0.825**	0.828
TwitterAAE	2.591	**2.453**	2.455
Dolma v1.5	0.876	0.804	**0.791**
Dolma 100 Subreddits	1.089	1.022	**1.001**
Manosphere	1.112	1.057	**1.030**
RedPajama	0.946	0.775	**0.737**
4chan	1.162	1.023	**0.952**
Dolma 100 PLs	1.106	0.694	**0.511**
Macro average	1.061	0.983	**0.972**

Table 1: **Our run at 10^{22} training FLOPs reaches a lower Paloma BPB than its rewriter on 10 of 16 sources.** Per-source Paloma BPB for the natural-only ($R = 0$) and predicted-optimum ($R = 0.32$) runs at 10^{22} training FLOPs and for SmolLM2-1.7B-Instruct, sorted by the predicted-optimum run's margin over the rewriter; the rule separates the sources where it is at or below the rewriter. Bold marks the lowest BPB per row. Appendix N.3 reports all four runs at 10^{22} training FLOPs and the SmolLM2 base model.

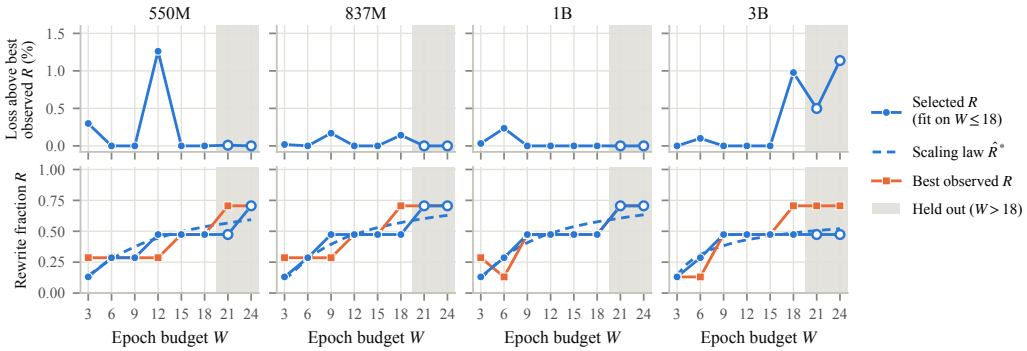

Figure 4: **The $\widehat{R}^*$ selected by the scaling law costs at most 1.14% in loss at held-out budgets.** The law (Eq. 1) is fit per target size to budgets through $W = 18$; $W = 21$ and $W = 24$ are held out (shaded, open markers). **Top:** measured final-checkpoint Paloma macro BPB at $\widehat{R}^*$, as a percentage above the loss at the best observed R on the same six-value grid. **Bottom:** $\widehat{R}^*$ (blue), the best observed R (orange), and the law's optimum over continuous $R \in [0, 1]$ (dashed).

At each budget, we sweep rewrite fractions $R \in \{0, 0.13, 0.29, 0.47, 0.71, 1\}$.[2] We report Paloma bits per byte at the final checkpoint (best-checkpoint results: Appendix H.2).

Rewriting methodology. For each target model, we pretrain a rewriter model with the same number of parameters on the seed set (i.e., $W = 1$; Appendix E). Pretrained models typically do not follow instructions, so we then fine-tune the rewriter on GLM-5.2 (Zhipu AI, 2026) rewrites of 1000 seed-set documents (7M tokens). We select the checkpoint with an LM judge and use it to rewrite the seed set under the same four FinePhrase prompts (Appendix F).[3] For synthetic rewrites within each mixture, we sample the four rewriting styles in proportion to their token counts.

3.3.1 Data-Limited Experimental Results

The best observed rewrite fraction increases in epoch budget and is predictable at unseen budgets (RQ1). The best observed rewrite fraction—the tested R resulting in the lowest final loss

[2]The rewrite fractions are chosen so that the fraction of FLOPs spent generating and training on rewrites, $5R/(3 + 2R)$, runs from 0 to 1 in steps of 0.2 (our rewriter has the same parameter count as our target model).

[3]The small set of rewritten documents limits possible knowledge transfer from GLM-5.2 to the rewriter. Appendix G compares this recipe with the SmolLM2 SFT+DPO recipe.

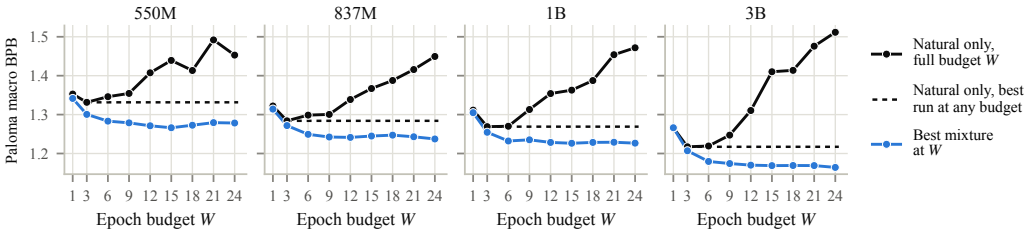

Figure 5: Final-checkpoint Paloma macro BPB against the epoch budget W for each target size. **Solid black:** natural-only epoching ($R = 0$) for the full budget of W epochs over the seed set. **Dotted black:** the best natural-only run at any budget, defined by the loss reached by stopping once further epochs raise it. **Blue:** best mixture at each W, the run with the lowest final loss at that budget over all rewrite fractions, so it coincides with the natural-only point wherever natural-only is best.

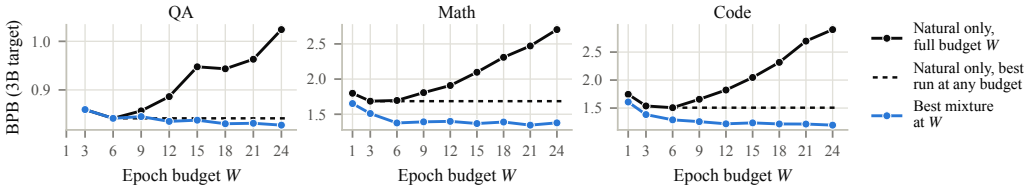

Figure 6: **Rewrites beat natural-only epoching on Math and Code at every budget, but on QA only after $W = 6$.** Final-checkpoint BPB for the 3B target against the epoch budget W on QA, Math, and Code, with the same three series as Figure 5. Appendix H.1 shows every target size.

at each size and budget—increases with the epoch budget W at every target size (Figure 4). For the 3B target, the best observed R increases from 0.13 at $W = 3$ to 0.71 at $W = 24$.

We predict $\widehat{R}^*$ in the data limited setting by fitting the rewriting scaling law (Eq. 1) for each target size to budgets through $W = 18$ and forecast $\widehat{R}^*$ for the held-out $W = 21$ and $W = 24$ runs. Figure 4 presents the results of this experiment. At R^*, the measured loss is within 0.74% of the forecast at every target size and both held-out budgets. For the 550M, 837M, and 1B target models, $\widehat{R}^*$ empirically minimizes Paloma BPB among tested R (except for 550M at $W = 21$, where it is within 0.01% of the Paloma BPB at the best choice of R). For the 3B target model at $W = 21$ the scaling law selects $\widehat{R}^* = 0.47$ and achieves a Paloma BPB of 1.175, 0.5% above the best observed $R = 0.71$; at $W = 24$ the scaling law selects $\widehat{R}^* = 0.47$ and achieves a Paloma BPB of 1.178, 1.14% above the loss at the best observed $R = 0.71$.

Rewrites reach a lower loss than natural-only epoching at every model size (RQ2). Figure 5 compares natural-only ($R = 0$) models to those trained at $\widehat{R}^*$ across a range of epoch budgets W; in particular, these plots show the final checkpoint's Paloma BPB, with the result that for epoch budgets $W > 3$ the natural-only runs overfit. Training at $\widehat{R}^*$ results in lower Paloma BPB than training at $R = 0$ at every size.

The best observed rewrite fraction depends on the task. Figure 6 repeats Figure 5 for the 3B target on QA, Math, and Code. On Math and Code the best mixture with $R > 0$ beats the best natural-only run at every budget. On QA the natural-only run is best through $W = 6$, by 0.014 BPB; beyond that, including synthetic rewrites outperforms.

4 Related Work

Data-constrained pretraining. When the corpus is finite, additional compute means additional epochs, whose returns diminish (Muennighoff et al., 2023). Lovelace et al. (2026) model this with a repetition penalty that activates beyond the first epoch, which we adopt as the $F(R, D, T)$ term of

Eq. (1). Kim et al. (2025; 2026) study the same regime from the perspective of a fixed data budget and unlimited compute, which is our data-limited setting.

Synthetic rewriting. Rewriting prompts a language model to restate a pretraining document, typically in a target style, and trains on the result alongside or instead of the original (Maini et al., 2024). Rewrites are now a component of several high-quality pretraining corpora (Su et al., 2024; Nguyen et al., 2025; Niklaus et al., 2026; Mahabad et al., 2025; Fujii et al., 2025; Ben Allal et al., 2025), and Maini et al. (2025) report that their benefit persists with scale.

Rewriting as distillation. Whether rewrites help because the rewriter knows more than the corpus is a standing question. Maini et al. (2024) ablate the rewriter and find that a 1.8B model retains most of the gain of a 7B one. Maini et al. (2025) find at trillion-token scale that returns to rewriter size saturate near 3B parameters and that a rewriter's benchmark scores are a poor guide to the value of its rewrites. Yang et al. (2025b) remove the external rewriter altogether, training a synthesizer on the pretraining corpus itself, and show that a 3B model still gains from the documents it synthesizes. Section 3.2 and Section 3.3 measure whether the rewriter's BPB caps the gains (RQ2) in each setting.

Scaling laws for synthetic pretraining data. Three recent studies fit scaling laws to synthetic pretraining data. Kang et al. (2025) train over a thousand models of 100M–3B parameters on mixtures of CommonCrawl with rephrased and textbook-style synthetic data from a 7B generator, fitting a separate Chinchilla law to each mixture: the best rephrased mixtures lie near 30% synthetic, pure rephrased data underperforms natural text, and mixtures containing rephrased data are projected to reach a lower irreducible loss than natural text alone. Qin et al. (2026) unify compute- and data-optimal scaling through a token-effectiveness function that discounts derived tokens, repetitions or SmolLM2-1.7B-instruct paraphrases of a fixed fresh corpus, relative to fresh ones, fit jointly on models of 14M–600M parameters. They find that effectiveness falls with model size and tokens per parameter. Qin et al. (2025) scale synthetic math data alone, generated from FineWeb-Edu math documents by a 123B model, in continued training of 1B–8B base models, and find a rectified scaling law in synthetic tokens that plateaus near 300B tokens, with larger models needing fewer. None of the three account for compute spent on generation, so their optimal mixtures answer a different question: how much to rewrite when rewrites are free. Qin et al. (2026) project that paraphrasing should stop helping beyond 7B parameters. However, our 22B-parameter run at the predicted optimum reaches 0.983 Paloma BPB against 1.061 for natural data alone at the same budget. All three also use an off-the-shelf generator stronger than the models they train. In the data-limited setting we pretrain the rewriter on the seed set and instruction-tune it on only 7M tokens of external rewrites, so gains from its rewrites can draw on little knowledge the corpus lacks. Appendix N.1.1 tests the token-effectiveness model of Qin et al. (2026) against ours.

5 Conclusion and Limitations

We asked whether scaling laws predict the cost-adjusted optimal rewrite fraction and associated loss (RQ1) and whether rewrites help when targeting capabilities beyond the rewriter (RQ2). Our rewriting scaling law answers both. In the compute-limited setting, the rewrite fraction fit at small budgets minimizes loss among the fractions we train at 33× the compute, and a pretrained model with 32% rewritten training data outperforms the rewriter on 10/16 general web text domains. In the data-limited setting, we similarly find that the optimal rewrite fractions and losses of held-out budgets can be predicted via scaling laws, and that a positive proportion of rewrites is additive even when the rewriter model is less capable than the target model.

Limitations. Several choices limit the extent of our results. One is the choice of the natural seed set: every optimal rewrite fraction we report is for FineWeb-Edu. We also study this question on the DCLM seed set, where rewrites still help but by less (Appendix N.7), suggesting that a strong natural seed set benefits less from rewriting. We also test only the four FinePhrase rewrite styles and one external rewriter in each setting, so we cannot say how the optimal fraction varies across rewrite prompt distributions or across a family of rewriters. Finally, our compute accounting is approximate: we charge $4N_r$ FLOPs per rewritten token, which approximates decoding as costing the same FLOPs as prefill and assumes the rewrite is the same length as its source; we do not account for costs associated with pretraining the rewriter model.

AI USE STATEMENT

This paper studies scaling laws for synthetic data generated by language models. Rewrites in the compute-limited setting are FinePhrase rewrites generated by SmolLM2-1.7B, and rewrites in the data-limited setting are generated by rewriters we pretrain ourselves and SFT on GLM-5.2 rewrites of 1000 seed-set documents. Finally, an LM judge selects the rewriter checkpoint (Section 3.1, Appendices F and J). We used AI assistants to discuss analysis choices, to implement and debug analysis code, for initial drafts of paper sections, and for feedback on prose. We have reviewed, verified, and updated code and text originating from AI.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. MultiPL-E: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 2023. URL https://arxiv.org/abs/2208.08227.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2305.12524.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Allyson Ettinger et al. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Kazuki Fujii, Yukito Tajima, Sakae Mizuki, Masaki Kawamura, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Masanari Oi, Taishi Nakamura, Takumi Okamoto, Shigeki Ishida, Kakeru Hattori, Youmi Ma, Hiroya Takamura, Rio Yokota, Jun Sakuma, and Naoaki Okazaki. Rewriting pretraining data boosts llm performance in math and code, 2025. URL https://arxiv.org/abs/2505.02881.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL https://arxiv.org/abs/2101.00027.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smoth-

ers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

David Hall, Percy Liang, and Marin Community. Levanter: Legible, scalable, reproducible foundation models with JAX. https://github.com/marin-community/levanter, 2023. Software.

Will Held. Scaling laws that extrapolate 300× past the fit. https://www.openathena.ai/blog/delphi/, may 2026. Open Athena Blog.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021b. URL https://arxiv.org/abs/2103.03874.

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, 2022. URL https://arxiv.org/abs/2203.15556.

Feiyang Kang, Newsha Ardalani, Michael Kuchnik, Youssef Emad, Mostafa Elhoushi, Shubhabrata Sengupta, Shang-Wen Li, Ramya Raghavendra, Ruoxi Jia, and Carole-Jean Wu. Demystifying synthetic data in llm pre-training: A systematic study of scaling laws, benefits, and pitfalls, 2025. URL https://arxiv.org/abs/2510.01631.

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, 2020. URL https://arxiv.org/abs/2001.08361.

Konwoo Kim, Suhas Kotha, Percy Liang, and Tatsunori Hashimoto. Pre-training under infinite compute, 2025. URL https://arxiv.org/abs/2509.14786.

Konwoo Kim, Suhas Kotha, Yejin Choi, Tatsunori Hashimoto, Nick Haber, and Percy Liang. Data-efficient pre-training by scaling synthetic megadocs, 2026. URL https://arxiv.org/abs/2603.18534.

Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. The depth-to-width interplay in self-attention, 2021. URL https://arxiv.org/abs/2006.12467.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2206.14858.

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, 2024. URL https://arxiv.org/abs/2406.11794.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, Jo ao Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Mu noz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder: may the source be with you!, 2023. URL https://arxiv.org/abs/2305.06161.

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

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groeneveld, Iz Beltagy, Hannaneh Hajishirzi, Noah A. Smith, Kyle Richardson, and Jesse Dodge. Paloma: A benchmark for evaluating language model fit. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=yjj8ele147.

Rabeeh Karimi Mahabad, Sanjeev Satheesh, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc-math: A 133 billion-token-scale high quality math pretraining dataset, 2025. URL https://arxiv.org/abs/2508.15096.

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, 2024. URL https://arxiv.org/abs/2401.16380.

Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, et al. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining, 2025. URL https://arxiv.org/abs/2508.10975.

MiniMax. MiniMax-M2.7. https://www.minimax.io/news/minimax-m27-en, 2026.

Bruno Mlodozeniec, Pierre Ablin, Louis Béthune, Dan Busbridge, Michal Klein, Jason Ramapuram, and Marco Cuturi. Completed hyperparameter transfer across modules, width, depth, batch and duration. In *ICLR*, 2026. URL https://arxiv.org/abs/2512.22382.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models, 2023. URL https://arxiv.org/abs/2305.16264.

Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. Recycling the web: A method to enhance pre-training data quality and quantity for language models, 2025. URL https://arxiv.org/abs/2506.04689.

Joel Niklaus, Atsuki Yamaguchi, Michal Štefánik, Guilherme Penedo, Hynek Kydlíček, Elie Bakouch, Lewis Tunstall, Edward Emanuel Beeching, Thibaud Frere, Colin Raffel, Leandro von Werra, and Thomas Wolf. How can we synthesize high-quality pretraining data? a systematic study of prompt design, generator model, and source data, 2026. URL https://arxiv.org/abs/2604.13977.

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, 2024. URL https://arxiv.org/abs/2406.17557.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In *Proceedings of Machine Learning and Systems*, volume 5, pp. 606–624, 2023. URL https://arxiv.org/abs/2211.05102.

Tian Qin, Kimia Hamidieh, and David Alvarez-Melis. Bridging compute- and data-optimal pretraining, 2026. URL https://arxiv.org/abs/2607.25271.

Zeyu Qin, Qingxiu Dong, Xingxing Zhang, Li Dong, Xiaolong Huang, Ziyi Yang, MAHMOUD KHADEMI, Dongdong Zhang, Hany Hassan Awadalla, Yi R. Fung, Weizhu Chen, Minhao Cheng, and Furu Wei. Scaling laws of synthetic data for language model. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=UmUXPXHtdl.

Nicholas Roberts, Niladri Chatterji, Sharan Narang, Mike Lewis, and Dieuwke Hupkes. Compute optimal scaling of skills: Knowledge vs reasoning, 2025. URL https://arxiv.org/abs/2503.10061.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset, 2024. URL https://arxiv.org/abs/2412.02595.

Joshua Vendrow, Edward Vendrow, Sara Beery, and Aleksander Madry. Do large language model benchmarks test reliability?, 2025. URL https://arxiv.org/abs/2502.03461.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of llm scaling based on human-generated data, 2022. URL https://arxiv.org/abs/2211.04325.

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them ii: Hyperball optimization, 2026. URL https://arxiv.org/abs/2606.16899.

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

Zitong Yang, Aonan Zhang, Hong Liu, Tatsunori Hashimoto, Emmanuel Candès, Chong Wang, and Ruoming Pang. Synthetic bootstrapped pretraining. In *ICLR*, 2025b. URL https://www.arxiv.org/abs/2509.15248.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://arxiv.org/abs/1905.07830.

Zhipu AI. GLM-5.2. https://github.com/zai-org, 2026.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023. URL https://arxiv.org/abs/2305.11206.

APPENDIX ROADMAP

Setup and evaluation
- Data sources (page 16)
- Model architecture (page 16)
- Hyper-parameter recipe (page 16)
- Evaluation tasks and subtasks (page 17)
- Mixture construction (page 22)

Data-limited experiments
- Rewriter capacity (page 18)
- Checkpoint selection with an LM judge (page 19)
- Alignment ablation (page 20)
- Complete budget grid (page 20)
- Full-budget U-curves (Figure 11, page 23)
- Domain loss across target sizes (Appendix H.1, page 22)
- Best-checkpoint evaluation (page 22)

Rewrite analysis
- Rewrite generation and cost (page 23)
- Style and line-content ablations (page 23)
- Qualitative rewrite examples (page 47)
- Rewriting scaling law details (page 25)
- Allocation dependence on R (page 27)

Compute-limited experiments
- Experimental grid (page 28)
- Full-R IsoFLOP surface (Figure 14, page 29)
- Per-R frontier and checkpoint check (Figures 15–16, page 30)
- Infinite-compute extrapolation (page 28)
- Fits and forward forecasts (page 32)
- Curvature ablation (page 35)
- Domain-specific optima (page 36)
- Evaluations of the 10^{22}-FLOP runs (page 38)
- Break-even allocations (page 39)
- Loss- versus accuracy-based fits (page 40)
- Surface-specification ablations (page 42)

DCLM diagnostics
- DCLM IsoFLOP ladders (Figure 28, page 45)
- DCLM vs. FineWeb-Edu ($R = 0$) (Figure 29, page 46)
- Source quality: DCLM rewrite sweep and transfer (page 42)

A DATA SOURCES

Natural text. The primary experiments use FineWeb-Edu (Penedo et al., 2024). We drop every document whose WARC file is dated 2023 or later, to limit LM-generated text in the natural corpus, and shuffle the remaining documents. The data-limited experiments fix a 5B-token seed set from this corpus. The compute-limited experiments sample fresh documents at each training-token budget. The source-quality control replaces FineWeb-Edu with DCLM (Li et al., 2024).

Rewritten text. The compute-limited experiments draw rewrites from FinePhrase (Niklaus et al., 2026), which contains approximately 338M rewrites of FineWeb-Edu documents in each of the FAQ, math, table, and tutorial styles. The data-limited experiments generate rewrites of the seed set with size-matched bootstrapped rewriters (Appendix J).

B MODEL ARCHITECTURE

All runs use the dense Qwen3 architecture, the Llama 3 tokenizer (Grattafiori et al., 2024), and a sequence length of 4096. The hidden dimension d determines each model size. The MLP intermediate dimension is $4d$. Attention heads have dimension 128, giving $d/128$ query and key–value heads. The log-corrected depth-to-width ratio of Levine et al. (2021) sets the number of layers:

$$L(d) = \text{round}\left(\frac{d}{55 + 4\log_2 d}\right). \tag{4}$$

This rule assigns 11 layers at $d = 1024$ and 21 layers at $d = 2048$. Appendix C specifies the candidate widths and selection rule.

C HYPER-PARAMETER TUNING

The Completed(d)P recipe of Mlodozeniec et al. (2026), implemented in Marin (Held, 2026), scales one reference configuration to each run's batch size B_{batch} and training-token count D. We measure B_{batch} in sequences of 4096 tokens.

Optimizer and schedule. We train with Adam and Hyperball (Wen et al., 2026); Wen et al. (2026) show that Hyperball makes the optimal learning rate robust across model sizes. Following Held (2026), we tune the learning rates on the 156M-parameter reference model with Paloma as the objective, then scale the learning rates and the remaining hyper-parameters to each run's batch size and token count. Every run uses linear warm-up over the first 10% of steps and linear decay to zero.

Reference point. The reference configuration is a $d = 512$ (156M-parameter) model trained with $B_{\mathrm{batch},0} = 64$ and $D_0 = 2.5 \times 10^9$ tokens on FineWeb-Edu. Its hyper-parameters are the best of a Gaussian-process search over 430 configurations at this scale, selected on Paloma validation macro cross-entropy as in Delphi (Held, 2026): Hyperball learning rate $\eta_0 = 6.30 \times 10^{-3}$, Adam learning rate $\eta_0^{\mathrm{A}} = 6.56 \times 10^{-4}$, $\epsilon_0 = 1.85 \times 10^{-8}$, $\beta_1 = 0.9$, and $\beta_{2,0} = 0.9999$.

Scaling rules. With $r/r_0 = (B_{\mathrm{batch}}\, D_0)/(B_{\mathrm{batch},0}\, D)$, a run with batch size B_{batch} and D tokens uses

$$\eta = \min\left(0.01,\ \eta_0 \sqrt{B_{\mathrm{batch}}/B_{\mathrm{batch},0}}\,(D_0/D)^{0.3}\right), \tag{5}$$

$$\eta^{\mathrm{A}} = \min\left(0.01,\ \eta_0^{\mathrm{A}} \sqrt{r/r_0}\right), \tag{6}$$

$$\epsilon = \epsilon_0 \sqrt{r_0/r}, \tag{7}$$

$$\beta_2 = \mathrm{clip}\left(\beta_{2,0}^{B_{\mathrm{batch}}/B_{\mathrm{batch},0}},\ 0.9,\ 0.9999\right). \tag{8}$$

The β_2 rule keeps its half-life constant in tokens. Every run uses gradient clipping at 0.1, z-loss weight 10^{-7}, linear warm-up over the first 10% of steps, and linear decay to zero.

Batch size. Each run targets 2^{16} steps. We round $D/(2^{16} \cdot 4096)$ up to a power of two, then halve the batch size until $\eta < 0.01$, $\eta^{\mathrm{A}} < 0.01$, and $\beta_2 > 0.9$, subject to $8 \le B_{\mathrm{batch}} \le 8192$.

Candidate models per budget. For each training budget C, we enumerate hidden dimensions from 512 in increments of 128 when $C \le 2 \times 10^{19}$ FLOPs and 256 otherwise. Each architecture follows Appendix B and receives $D = C/(3F_{\mathrm{fwd}})$ tokens, where F_{fwd} denotes forward FLOPs per token. Budgets are labeled by this nominal C; the scaling-law fits and all compute accounting use $C = 6ND$ as in Section 2. We exclude models above $12\mathrm{B} \cdot \max\left(1, \sqrt{C/(3 \times 10^{20})}\right)$ parameters or 250 tokens per parameter.

D Evaluation Details

We evaluate language-model fit with Paloma (Magnusson et al., 2024) and question answering, mathematics, and code with offline evaluation harnesses.[4] Unless stated otherwise, each metric is bits per byte (BPB): reference-continuation negative log-likelihood in bits divided by UTF-8 byte length. Lower is better.

Paloma. Paloma macro BPB averages 16 sources uniformly: 4chan, C4 100 Domains, C4 English, Dolma v1.5, Dolma 100 Programming Languages, Dolma 100 Subreddits, Falcon RefinedWeb, Gab, M2D2 S2ORC, M2D2 Wikipedia, Manosphere, mC4, Penn Treebank, RedPajama, TwitterAAE, and WikiText-103. Both settings use the training-loop evaluator at the models' trained 4096-token context and pool bytes within each source. We evaluate SmolLM2 at a forced 4096-token context and match the within-source pooling for the rewriter comparison.

Question answering. The QA suite contains ARC-Easy, ARC-Challenge, HellaSwag, and all 57 MMLU subjects. We first average the two ARC tasks and the 57 MMLU subjects within their families, then average ARC, HellaSwag, and MMLU uniformly. The MMLU subjects are:

[4]The SmolLM2 Paloma reference uses OLMES (`https://github.com/allenai/olmes`). The domain-likelihood suites use OLMo Eval (`https://github.com/allenai/olmo-eval`).

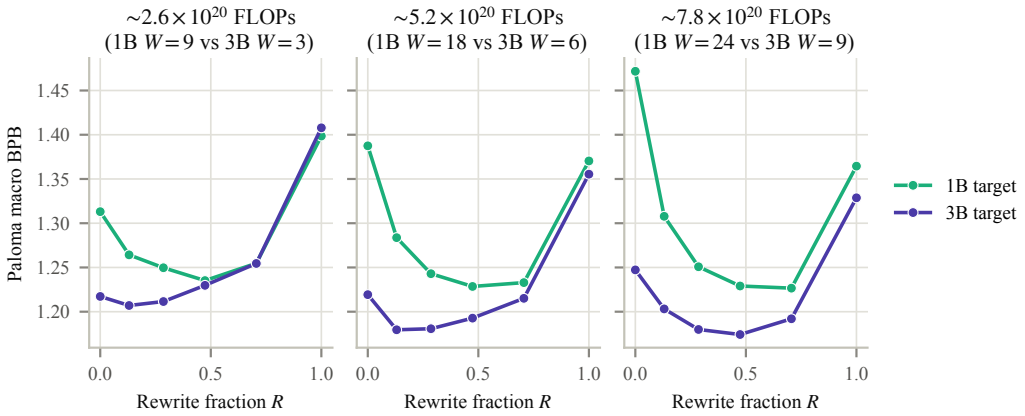

Figure 7: 1B and 3B targets at matched total compute. Each panel pairs the nearest budgets on the grid, with FLOP ratios 0.97, 0.97, and 1.09.

- *STEM:* abstract algebra, anatomy, astronomy, college biology, college chemistry, college computer science, college mathematics, college physics, computer security, conceptual physics, electrical engineering, elementary mathematics, high-school biology, high-school chemistry, high-school computer science, high-school mathematics, high-school physics, high-school statistics, machine learning, and medical genetics;
- *humanities:* formal logic, high-school European history, high-school US history, high-school world history, international law, jurisprudence, logical fallacies, moral disputes, moral scenarios, philosophy, prehistory, professional law, and world religions;
- *social sciences:* econometrics, high-school geography, high-school government and politics, high-school macroeconomics, high-school microeconomics, high-school psychology, human sexuality, professional psychology, public relations, security studies, sociology, and US foreign policy; and
- *other:* business ethics, clinical knowledge, global facts, human aging, management, marketing, college medicine, miscellaneous, nutrition, professional accounting, professional medicine, and virology.

Rank classification scores every answer choice. The scaling surfaces use gold-answer BPB (`qa_rc:bpb:base_easy`). The secondary metric is the fraction of examples on which the gold answer ranks first (`qa_rc:base_easy`).

Mathematics. The Math metric `math_bpb:all` uniformly averages nine gold-answer BPB tasks: GSM8K-Platinum, TheoremQA, and the seven Minerva MATH subjects (algebra, counting and probability, geometry, intermediate algebra, number theory, prealgebra, and precalculus).

Code. The Code suite contains three-shot HumanEval, three-shot MBPP, and three-shot multilingual MBPP-v2fix in Bash, C, C++, C#, Go, Haskell, Java, JavaScript, MATLAB, PHP, Python, R, Ruby, Rust, Scala, Swift, and TypeScript. We first average the 17 multilingual variants, then average HumanEval, MBPP, and multilingual MBPP uniformly (`code_bpb:base_easy`). The metric is reference-code BPB, not execution accuracy.

E Data Limited Base Rewriters

Effect of rewriter capacity. The capacity control trains the 1B target on rewrites from the 550M and 3B rewriters at $W \leq 12$. At $W = 12$, the 550M rewriter sits 0.011 BPB below the 3B rewriter at $R = 0.13$ and trails it by 0.069 BPB at $R = 1$ (Figure 8, left). Training the 1B target on 3B rewrites closes 52% of the 1B–3B target gap at $R = 1$ and $W = 12$, and between -4% and 22% at

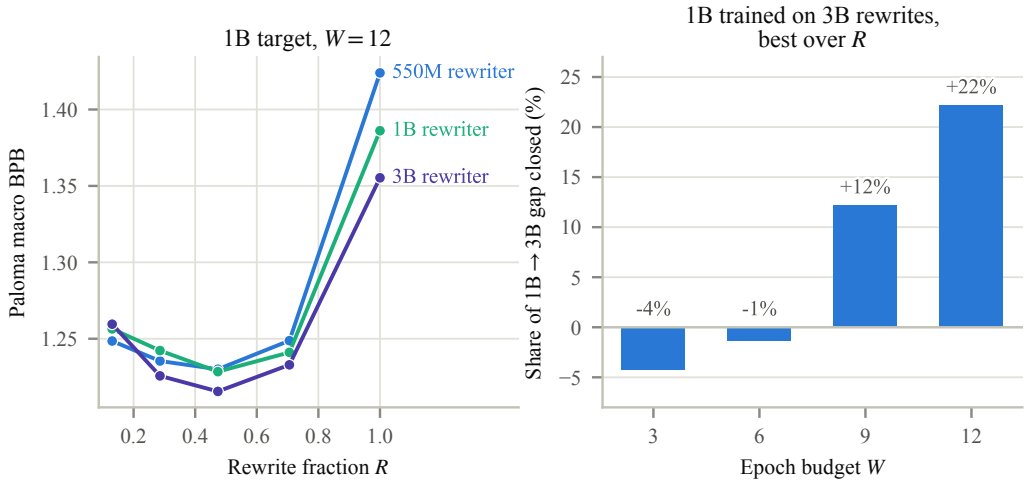

Figure 8: Rewriter capacity. **Left:** 1B target loss at $W = 12$ with rewrites from the 550M, 1B, and 3B rewriters. **Right:** share of the 1B–3B target gap closed by training the 1B target on 3B rewrites, at each budget's best mixture.

each budget's best mixture (right). Figure 7 pairs the 1B and 3B targets at the nearest total-compute budgets on the grid. The 3B target has lower BPB in 17 of the 18 pairs, by 0.031 BPB on average at the smallest budget pair and 0.088 BPB at the largest.

F Rewriter Instruction Tuning

Each data-limited rewriter receives supervised fine-tuning (SFT) after pretraining on the seed set. The SFT set contains 4000 examples: GLM-5.2 (Zhipu AI, 2026) rewrites of 1000 seed-set documents under each of the four FinePhrase prompts, 7M tokens in total, at sequence length 8192. Following Zhou et al. (2023), we train with a batch size of 32 examples and linear decay of the learning rate to zero, with no warm-up. We sweep the learning rate and the number of epochs per rewriter size and select the checkpoint with the LM judge of Appendix F.1.

F.1 Checkpoint Selection with an LM Judge

The in-house rewriter generates the synthetic data for every data-limited run (Section 3.3), so we choose its fine-tuning learning rate and number of epochs by the quality of the rewrites each checkpoint generates. At 550M and 837M the checkpoints are the 12 from learning rates 10^{-5} to 5×10^{-4} at 10 or 15 epochs, at 3B the six from 5×10^{-5} to 2×10^{-4}, and at 1B a shortlist of seven of 31 fine-tuning runs that also covers 5 epochs (Table 2). Every checkpoint rewrites the same 100 randomly sampled FineWeb-Edu documents under the four FinePhrase prompts, 400 rewrites per checkpoint. The reference for each document and prompt is its FinePhrase rewrite, generated by SmolLM2-1.7B, the rewriter of the compute-limited setting.

GLM-5.2 (Zhipu AI, 2026) and MiniMax-M2.7 (MiniMax, 2026) judge pairs of rewrites of the same document under the same prompt. A pair is either a checkpoint's rewrite and the SmolLM2-1.7B rewrite or the rewrites of two checkpoints, and not every pair is judged. A judge receives the source document, the rewrite prompt, and the two rewrites, and returns $+1$ if the first rewrite is better, -1 if the second is better, or 0 for a tie (prompt below). Each pair is judged in both presentation orders. Elo ratings computed from each judge's preferences place every checkpoint and SmolLM2-1.7B on one scale, and the rating difference gives each checkpoint's expected win rate against SmolLM2-1.7B. A checkpoint's score is this win rate averaged over the two judges.

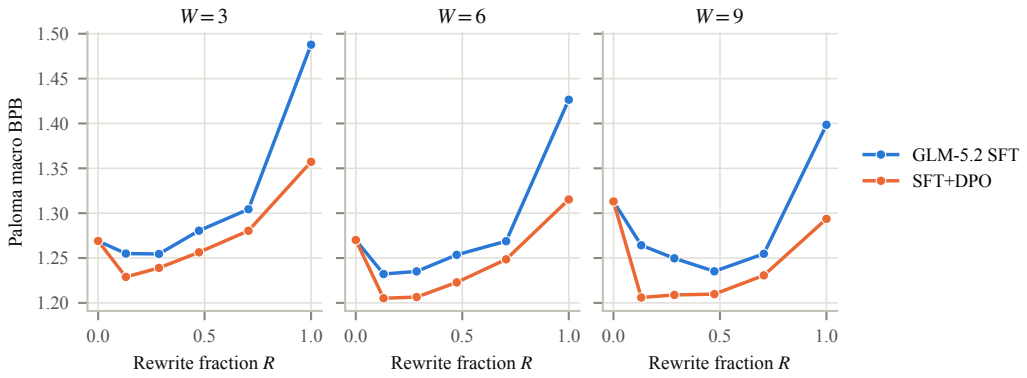

Figure 9: Final-checkpoint Paloma macro BPB for the 1B target trained on rewrites from the SFT and SFT+DPO rewriters at $W \in \{3, 6, 9\}$. Both conditions share the natural-only $R = 0$ runs.

Judge prompt

Compare how well the two rewrites rewrite the source document according to the style specified in the rewrite prompt.

Source document: {`source`}

Rewrite prompt: {`prompt`}

Rewrite A: {`rewrite_a`}

Rewrite B: {`rewrite_b`}

Return +1 if A is better, -1 if B is better, or 0 for a tie.

Table 2 ranks the checkpoints by score and marks the checkpoint that generates the seed-set rewrites at each size: a learning rate of 10^{-4} for 15 epochs at 550M and 1B, 2×10^{-4} for 10 epochs at 837M, and 5×10^{-5} for 15 epochs at 3B. This is the highest-scoring checkpoint at 550M, 837M, and 3B. At 1B it scores 0.7 points below the highest-scoring checkpoint, and the confidence intervals of the two overlap. No checkpoint reaches a 50% win rate against SmolLM2-1.7B.

G Alignment

The data-limited rewriters receive supervised fine-tuning (SFT) on GLM-5.2 rewrites and LM-judge checkpoint selection (Appendix F). The alignment control instead follows the SmolLM2 post-training procedure (Ben Allal et al., 2025): SFT followed by direct preference optimization (DPO) on the 1B rewriter, with no LM-judge selection. Across the 15 runs with $R > 0$ at $W \in \{3, 6, 9\}$, SFT+DPO lowers Paloma macro BPB in all 15 and by 0.046 BPB on average (Figure 9). The programming-languages and RedPajama sources account for 80% of the reduction, and BPB increases on 8 of the 16 Paloma sources (Figure 10).

H Data-Limited Experimental Results

Figure 11 reports the full data-limited grid: four target sizes (550M, 837M, 1B, and 3B parameters), eight epoch budgets $W \in \{3, 6, 9, 12, 15, 18, 21, 24\}$, and six rewrite fractions $R \in \{0, 0.13, 0.29, 0.47, 0.71, 1\}$. Table 3 gives the total compute, accounting for rewrites, for every size and budget. The rewriter generates every rewrite token a run trains on, and no rewrite token is trained on twice, so the repetition penalty in Eq. (14) applies to the natural seed set only. Figure 4 in the main text shows $W \in \{3, 6, 12, 24\}$.

Size	SFT learning rate	Epochs	Mean win rate (%)
550M	10^{-4}	15†	28.3
	2×10^{-4}	15	27.7
	2×10^{-4}	10	26.6
	10^{-4}	10	25.0
	2×10^{-5}	15	24.8
	2×10^{-5}	10	23.8
	5×10^{-5}	10	23.2
	5×10^{-4}	10	22.6
	5×10^{-4}	15	22.3
	5×10^{-5}	15	21.9
	10^{-5}	15	19.9
	10^{-5}	10	19.4
837M	2×10^{-4}	10†	32.5
	2×10^{-4}	15	31.5
	10^{-4}	15	30.9
	10^{-4}	10	29.7
	5×10^{-5}	15	28.0
	2×10^{-5}	10	26.4
	5×10^{-5}	10	26.3
	2×10^{-5}	15	25.3
	5×10^{-4}	15	25.0
	5×10^{-4}	10	24.4
	10^{-5}	15	23.2
	10^{-5}	10	20.9
1B	2×10^{-4}	10	33.3
	10^{-4}	15†	32.6
	10^{-4}	10	31.7
	2×10^{-4}	5	31.7
	10^{-4}	5	30.7
	2×10^{-4}	15	30.6
	5×10^{-5}	15	30.3
3B	5×10^{-5}	15†	38.8
	10^{-4}	10	38.5
	5×10^{-5}	10	38.4
	10^{-4}	15	37.4
	2×10^{-4}	10	34.5
	2×10^{-4}	15	33.9

Table 2: **Win rates against the SmolLM2-1.7B FinePhrase rewrite, averaged over both judges.** Each row is a rewriter checkpoint. Scores are the Elo-derived win rate against the SmolLM2-1.7B rewrite of the same document and prompt, averaged over GLM-5.2 and MiniMax-M2.7. † marks the checkpoint that generates the seed-set rewrites. Rows are ordered by score within each size. The 1B results cover seven shortlisted configurations.

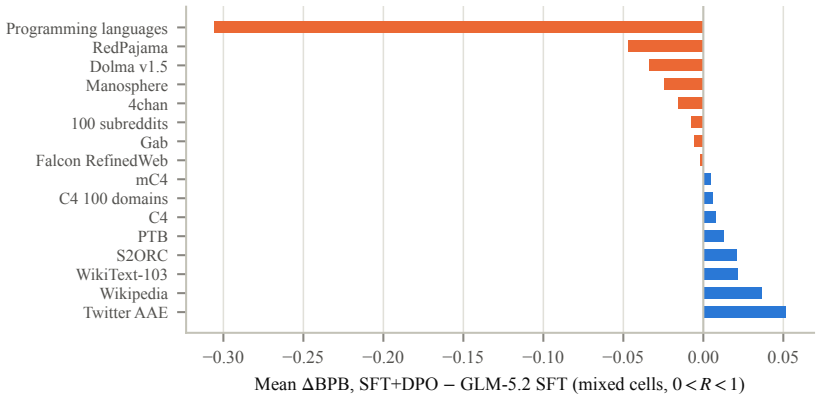

Figure 10: Mean Paloma BPB change per source from replacing the SFT rewriter with the SFT+DPO rewriter across the 16 mixed runs ($0 < R < 1$, $W \leq 12$). Orange bars denote lower BPB under SFT+DPO and blue bars lower BPB under the SFT rewriter, matching the colors of Figure 9.

Table 3: Total compute, accounting for rewrites, for the complete data-limited grid, in units of 10^{20} FLOPs. Each compute value has six rewrite-fraction runs.

Target size	Epoch budget W							
	3	6	9	12	15	18	21	24
550M	0.495	0.991	1.486	1.981	2.476	2.972	3.467	3.962
837M	0.753	1.507	2.260	3.013	3.766	4.520	5.273	6.026
1B	0.898	1.796	2.695	3.593	4.491	5.389	6.287	7.185
3B	2.606	5.211	7.817	10.422	13.028	15.633	18.239	20.844

H.1 Domain Loss Across Target Sizes

Figure 12 repeats Figure 6 for every target size. The best observed rewrite fraction on QA is 0 at $W \leq 6$ for the 3B target, at $W \leq 9$ for the 1B and 837M targets, and at every budget but one for the 550M target. On Math and Code it is interior at every size from $W = 3$. The QA, Math, and Code values come from our evaluation harness (Appendix D).

H.2 Best-Checkpoint Evaluation

Section 3.3 reports final checkpoints. The best-checkpoint analysis instead selects, for each run, the saved checkpoint with the lowest Paloma validation loss whenever it improves on the final checkpoint by at least 0.01 nats, and evaluates Paloma, QA, Math, and Code at that checkpoint. Paloma alone determines the selection. Selection changes the best observed rewrite fraction on Paloma in 4 of the 32 size–budget settings.

Figure 13 shows the 3B target at $W \in \{3, 6, 12, 24\}$. At $W = 24$, selection lowers natural-only Paloma BPB from 1.512 to 1.241, below pure rewriting at 1.264. The lowest Paloma BPB in the figure is at $W = 24$ and $R = 0.71$. QA and Code reach their minima at $W = 24$ and $R = 0.47$, and Math ties at $R = 0.13$ and $R = 0.47$ after rounding to two decimals.

I Mixture Construction

A mixture at rewrite fraction R draws its natural tokens from the filtered FineWeb-Edu documents (Appendix A) and its rewritten tokens from the FinePhrase rewrites of those documents, pooled across the math, FAQ, table, and tutorial styles and sampled by tokens. We train with Levanter (Hall et al., 2023) and build every mixture with its mixture dataset. The natural corpus and the rewrite

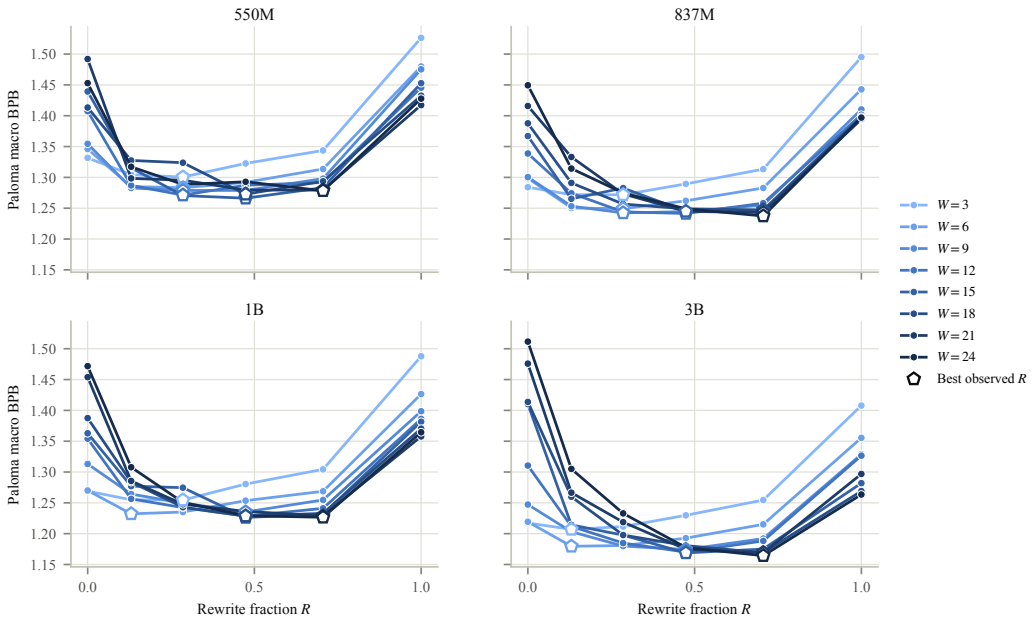

Figure 11: Final-checkpoint Paloma macro BPB for all 192 data-limited runs. Each panel shows one target size at all eight epoch budgets W and six rewrite fractions R. Open pentagons mark the best observed mixture at each size and budget.

pool enter as two components with weights $1 - R$ and R. Levanter partitions the training stream into fixed-size blocks and assigns the examples of each block to the two components in proportion to their weights, so every block-aligned prefix of the stream has rewrite fraction R up to rounding. The assignment is a deterministic function of the seed, so runs at the same R see the same interleaving.

J Generating Rewrites

Compute-limited generation. SmolLM2-1.7B generates the FinePhrase rewrites. We account for each rewritten training token as one input pass over its source document and one output pass over the rewrite, $4N_r$ FLOPs with $N_r = 1.71$B (Eq. 12).

Data-limited generation. Each target size uses a rewriter of the same size, pretrained on the seed set and instruction-tuned at sequence length 8192 on GLM-5.2 rewrites of 1000 seed-set documents under the four FinePhrase prompts (4000 examples). The rewriter then rewrites the seed set under the same prompts at temperature 1.0. Appendix L.2 converts the generated volume to the rewrite fraction R.

We generate the synthetic pool before target-model training by repeatedly sampling fresh rewrites of the same seed documents under the same four FinePhrase prompts at temperature 1.0. The pool remains fixed during training: we do not regenerate rewrites between training epochs. Each synthetic token used by a run is seen once, while natural tokens may be repeated.

K Rewrite-Style and Line-Content Ablations

We sample 6,000 documents from each FinePhrase variant and classify every line as prose, list, table, header, or code (Table 4). Table rewrites are 59% table lines. Tutorial rewrites have the largest code share, 0.80% of lines. This classification measures surface form and not faithfulness to the source document.

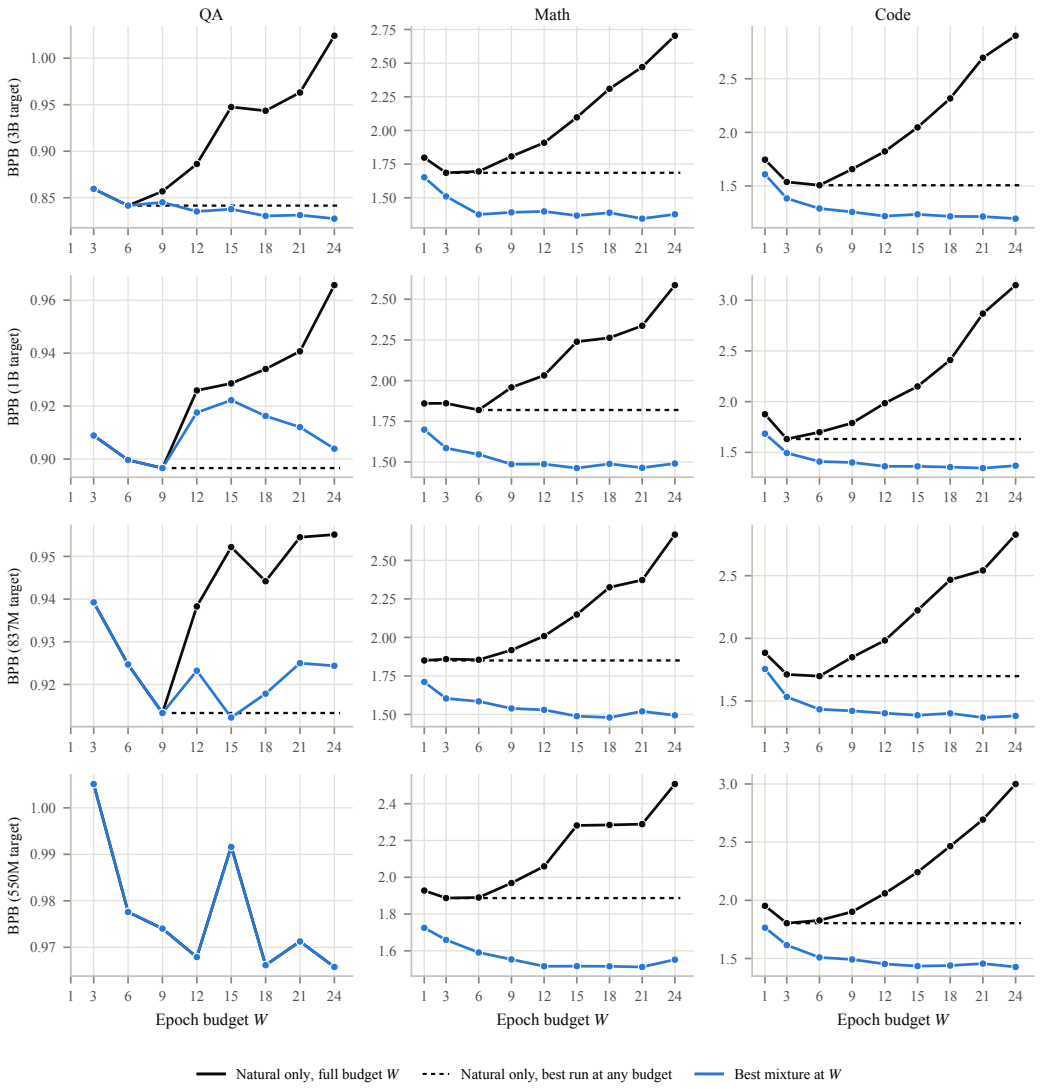

Figure 12: Final-checkpoint BPB against the epoch budget W for every target size (rows) on the QA, Math, and Code suites (columns). Series as in Figure 5.

Table 4: Line composition of 6,000 documents per FinePhrase variant, in percent of classified lines.

Variant	Prose	List	Table	Header	Code
Natural	84.41	14.32	1.26	0.01	0.00
FAQ	64.69	33.01	0.50	1.74	0.06
Math	69.02	30.06	0.19	0.28	0.45
Table	29.21	10.07	58.99	1.68	0.05
Tutorial	49.16	45.34	0.38	4.31	0.80

Style contrasts. The content ablation trains one run per arm and rewrite fraction, for 20 arms and 10 fractions. Four style-only arms keep one rewrite style, and four style-out arms remove one style. FAQ style-only has higher Paloma BPB than FAQ style-out at all 10 fractions, by 0.0226 BPB on average. Math style-only is higher at 7 of 10 fractions (+0.0032 BPB on average). Table style-only

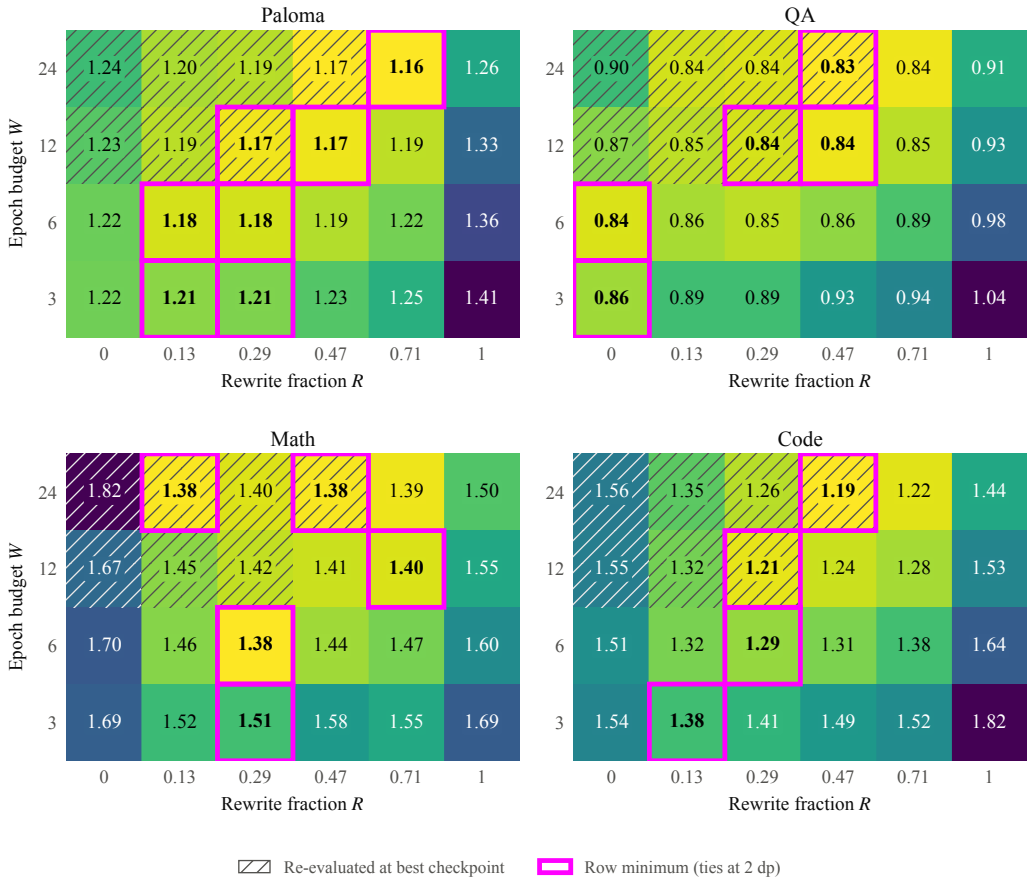

Figure 13: Best-checkpoint BPB for the 3B target by epoch budget W and rewrite fraction R. Hatching marks runs re-evaluated at an earlier checkpoint whose Paloma validation loss is at least 0.01 nats below the final checkpoint. Magenta outlines mark row minima, including ties after rounding to two decimals. Paloma alone selects each checkpoint.

is lower at all 10 fractions (-0.0039 BPB), and tutorial style-only is lower at 8 of 10 (-0.0038 BPB).

Line-type contrasts. Five arms remove one line type from every rewrite: code, headers, lists, prose, or tables. We compare each against the full mixture at the nine fractions below $R = 0.9$ and exclude the full-mixture run at $R = 0.9$. Every removal raises mean BPB, by 0.0120 to 0.0255 BPB, with the largest increase from removing prose.

L Rewriting Scaling Law Details

This section collects the derivations and the per-setting instantiations of the rewriting scaling law (Section 2), Eq. (1).

Mixture-dependent loss floor. We fit L in bits per byte; every statement below holds in any logarithmic base, with λ_n and λ_s in Eq. (10) rescaled accordingly. $E(R)$ is the irreducible loss of the mixture: the cross-entropy that a model with unlimited capacity and data would reach on the evaluation distribution when trained on a mixture of $(1 - R)$ natural and R rewritten data. We write p_e for the evaluation distribution, p_n and p_s for the natural and rewrite distributions, and $p_R = (1 - R)\, p_n + R\, p_s$ for the training mixture's distribution. A model with unlimited capacity

and data converges to p_R, so

$$E(R) = H(p_e) + \mathrm{KL}(p_e \,\|\, p_R)\,. \tag{9}$$

Note that only the KL divergence depends on R. Because $\mathrm{KL}(p \,\|\, q)$ is convex in q and p_R is linear in R, $E(R)$ is convex in R for every evaluation distribution. Evaluated on natural text ($p_e = p_n$) it is minimized at $R = 0$ and increases with R; evaluated on the rewrites themselves ($p_e = p_s$) the divergence falls to zero as $R \to 1$; evaluated on a distribution that neither source matches exactly, such as Paloma, the minimizer can be interior. A two-component approximation to the divergence (Appendix N.1.2) gives the parametric form

$$E(R) = E_0 - \lambda_n \log(1 - (1 - \rho_n)R) - \lambda_s \log(\rho_s + (1 - \rho_s)R)\,. \tag{10}$$

Appendix N.1.2 derives Eq. (10) and interprets $\lambda_n, \rho_n, \lambda_s, \rho_s$ as the share of the evaluation distribution each source covers better and the other source's relative density there. The first term is the cost of *removing* natural text and the second the benefit of *adding* rewrites. The minimizer $R^*_\infty = \arg\min_R E(R)$, the floor-optimal rewrite fraction, is unique because Eq. (10) is strictly convex for positive λ_n, λ_s, and is interior when $E'(0) < 0 < E'(1)$ (Appendix N.1.2).

Repetition penalty. $F(R, D, T) = P\,(1 - R)^s \left((1 - R)D/T - 1\right)_+$ is the repetition penalty of Lovelace et al. (2026), diluted by rewrites at rate s. It is zero when $(1 - R)D \le T$, as in every compute-limited run.

Rewrites shift the floor only. R enters Eq. (1) through $E(R)$ and, when $(1 - R)D > T$, through the dilution factor $(1 - R)^s$ and the natural epochs $(1 - R)D/T$ of the repetition penalty; it does not enter the capacity and data terms. This is the central modeling assumption of the paper: rewrites change *what* the model converges to, not *how fast* it gets there in N and D. It implies that the best loss at any training budget decomposes as

$$L^*(C, R) = L^*_{\mathrm{base}}(C) + \tilde{g}(R), \qquad \tilde{g}(R) = E(R) - E(0). \tag{11}$$

Here, $L^*(C, R) = \min_{6ND=C} L(N, D, R, T \to \infty)$ is the best loss over (N, D) at training budget C and $L^*_{\mathrm{base}}(C) \equiv L^*(C, 0)$. Figure 15 (Appendix N.1) tests how closely $N^*(C)$ and $D^*(C)$ remain stable across R.

Accounting for rewrites. Section 2 gives the generation cost of $4N_r$ FLOPs per rewritten training token for a rewriter with N_r parameters, so

$$C_{\mathrm{total}} = C + 4N_r RD, \qquad C = 6ND. \tag{12}$$

We do not account for the cost of training the rewriter as it is amortized across all experiments.

L.1 Compute-Limited Setting

Setup. The compute-limited setting reflects the non-frontier case in which natural data is plentiful and our goal is to characterize the strongest possible model subject to a fixed compute budget. We use an external rewriter: one that we do not train ourselves, is stronger than any model that we train, and is trained with more compute than our largest training run, and we do not account for its training cost. A mixture with rewrite fraction R draws a fraction R of its training tokens from rewrites and the remaining $1 - R$ from natural text. We fit the law to IsoFLOP sweeps: at each training budget C we train one grid of model sizes N with $D = C/(6N)$ tokens, and every rewrite fraction R uses the same grid.

Fitting the surface. We fit Eq. (1) with $F = 0$ to every run, since $(1 - R)D \le T$ for every compute-limited run, to obtain the floor $E(R)$, subtract the fitted shift $\tilde{g}(R)$ from every run, and fit the (N, D) backbone to the pooled runs with Chinchilla Approach 2 (Hoffmann et al., 2022). A quadratic in $\log D$ at each budget gives $D^*(C)$ and $L^*_{\mathrm{base}}(C)$, and power laws across budgets give α and β. We add an off-frontier curvature term that vanishes with zero slope on the frontier and leaves $N^*(C)$, $D^*(C)$, and $L^*_{\mathrm{base}}(C)$ unchanged. Appendix N.1.4 gives the exponent conversion, the fitted parameters, the leave-one-budget-out forecasts, and the curvature ablation.

Compute-optimal rewrite fraction. Under Eq. (11) the floor-optimal fraction R^*_∞ minimizes loss at any *training* budget. Accounting for generation changes the answer: at a fixed total budget, every rewritten token is paid for twice, once to generate and once to train on, so raising R shrinks the training budget available. Because natural data is plentiful in this setting, we assume an unlimited supply of natural tokens and define $L(N, D, R) = \lim_{T\to\infty} L(N, D, R, T)$. As $T \to \infty$ the repetition penalty $F(R, D, T)$ vanishes. The compute-optimal allocation minimizes this surface jointly over model size, token count, and rewrite fraction at a fixed total budget:

$$\left(N^*, D^*, \widehat{R}^*\right) = \underset{N,\, D,\, R\in[0,1]}{\arg\min} \; L(N, D, R) \quad \text{s.t.} \quad 6ND + 4N_r RD = C_{\text{total}}, \tag{13}$$

where we evaluate $L(N, D, R)$ with the off-frontier curvature term of Eq. (31) added, as in the fitted surface. Because each rewritten token also costs generation compute, the optimum with $R > 0$ trains a slightly larger model on fewer tokens than the natural-only allocation at the same training compute. The generation cost is linear in D while the training loss improves as a power law in compute, so the overhead of rewriting becomes relatively cheaper as budgets grow, and $\widehat{R}^*(C_{\text{total}})$ increases from near 0 toward $\widehat{R}^*_\infty$ (Figure 2).

L.2 Data-Limited Setting

Setup. The data-limited setting reflects the frontier case in which no stronger rewriter exists, we target a fixed model size, and the amount of natural data is fixed, but compute is the plentiful resource. A fixed seed set of T natural tokens and a target model with N parameters are given, and an internal rewriter of the same size, pretrained on the seed set, generates the rewrites. We do not account for the rewriter's training cost, which is shared across all data-limited experiments at that size. As above, budgets are marginal compute given the rewriter. As in the main paper, we measure the total compute budget by $W = C_{\text{total}}/(6NT)$. A run with D training tokens and rewrite fraction R trains on $(1 - R)D$ natural tokens and RD synthetic tokens, each synthetic token seen once. Training and generation together cost $6ND + 4NRD = 6NTW$, giving $D = 3TW/(3+2R)$. The run therefore passes over the natural seed set $3W(1 - R)/(3 + 2R)$ times.

Data-limited scaling model. With the target size fixed at N_0, the capacity term A/N_0^α in Eq. (1) is a constant and is absorbed into the floor. At epoch budget W and rewrite fraction R, the run sees $D = 3TW/(3 + 2R)$ tokens, counting natural and synthetic tokens alike: $3W(1 - R)/(3 + 2R)$ epochs over the seed set and the rest fresh rewrites, none repeated. Substituting into Eq. (1) gives

$$L_{\text{DE}}(R, W) = E(R) + \frac{B}{(3TW/(3 + 2R))^\beta} + P\,(1 - R)^s \left(\frac{3W(1 - R)}{3 + 2R} - 1\right)_+ . \tag{14}$$

At $R = 0$ the law reduces to the epoching law of Lovelace et al. (2026). At $R = 1$ the repetition penalty vanishes. In between, rewrites add fresh tokens and dilute repetition of the seed set. We fit the law separately for each target–rewriter pair, pooling all epoch budgets and rewrite fractions, because the size-matched rewriter changes the quality of the synthetic data with the target size (Figure 8). The minimizer of the fitted law over R is $\widehat{R}^*(W)$.

M Dependence of Efficient-Frontier Allocation on R

The shared surface fits one compute-optimal token count $D^*(C)$ across rewrite fractions. We test this restriction on the 583 runs of the five budgets with all eleven rewrite fractions. Both fits give every budget–fraction cell its own loss level. The restricted fit shares $D^*(C)$ across fractions. The alternative adds ten fraction-specific offsets to $\log_{10} D^*$, with $R = 0$ as the reference.

The offsets lower RMSE from 0.00831 to 0.00762 BPB, and the largest shift is $D^*(R)/D^*(0) = 1.196$ at $R = 0.8$. A wild bootstrap that resamples residual signs jointly within each architecture (400 draws) rejects equal D^* across fractions ($p = 1/401$). Figure 15 shows that a shift of this size does not change the selected model size, and Appendix N.6 compares richer surface specifications on the same runs.

N Compute-Limited Experimental Results

N.1 Scaling Model Details

The FineWeb-Edu ladders contain five training budgets from 3×10^{18} to 9×10^{19} FLOPs at eleven rewrite fractions $R \in \{0, 0.1, \ldots, 1\}$, and natural-only ($R = 0$) ladders at 1.8×10^{20} and 3×10^{20} FLOPs. The surface fit uses these 604 Paloma BPB observations and the 10^{22}-FLOP hero run ($R = 0.321$) as a frontier anchor. Nonzero-R runs at 1.8×10^{20} and 3×10^{20} FLOPs, the 10^{21} runs, and the remaining 10^{22} runs are validation runs outside the fit. The forward forecasts in Appendix N.1.4 refit the surface without each held-out budget.

The frontier location does not depend on R. The surface fit pools all rewrite fractions before locating the IsoFLOP optima (Appendix L.1), so the fitted $N^*(C)$ and $D^*(C)$ cannot vary with R by construction. Figure 15 tests the assumption by fitting a separate Approach-2 parabola in $\log D$ to each (R, C) sweep and taking its vertex. The 56 per-R optima over seven budgets lie within 0.85–1.28$\times$ of the shared $D^*(C)$ (RMS 10%), and per-R power laws give token exponents between 0.45 and 0.57 against the shared 0.46. The spread across R at a fixed budget matches the spread of the $R = 0$ ladder across budgets (0.85–1.03$\times$).

Figure 16 asks whether this spread changes the selected model size. The grid steps by 1.2–1.4$\times$ in N. In 23 of 54 sweeps the per-R vertex selects a neighboring grid size, never one two steps away. Most of these cases are at 1.8×10^{19} and 9×10^{19} FLOPs, where the shared N^* falls midway between two grid sizes. On each sweep's own parabola, the shared choice costs at most 0.0015 BPB (0.0002 BPB at the median), against the 0.082-BPB floor gain at $\widehat{R}^*_\infty$. The measured gap between the two grid runs (median 0.0015, maximum 0.0108 BPB) matches the parabola residuals and does not grow with R.

N.1.1 Do Rewrites Still Help in the Infinite-Compute Limit?

Figure 17 shows the measured evidence that rewrites shift the floor independently of scale. After subtracting each budget's $R = 0$ value, 82 IsoFLOP-optimal observations from nine budgets collapse onto one curve $\tilde{g}(R)$ with RMSE 0.0044 BPB.

The final fit gives $\widehat{R}^*_\infty = 0.328$ and a floor shift of -0.0825 BPB. Extrapolating both frontiers, a natural-only model at 10^{22} training FLOPs needs 5.01$\times$ as many tokens to match the estimated floor-optimal mixture, and the natural-only frontier stops reaching the mixture frontier at 3.61×10^{26} FLOPs (Figure 18). These magnitudes depend on the fit specification. An independent fit to all 671 runs, including the validation runs but holding three 10^{22}-FLOP runs out, gives a 12.17$\times$ multiplier at 10^{22} FLOPs with 95% bootstrap interval $[6.39, 45.01]$ and divergence at 1.75×10^{24} FLOPs. No run exceeds 10^{22} FLOPs, so both multipliers are extrapolations.

N.1.2 $E(R)$ Derivation

Let p_e denote the evaluation distribution, p_n the natural training distribution, and p_s the rewrite distribution. The mixture distribution is

$$p_R(x) = (1 - R)p_n(x) + Rp_s(x). \tag{15}$$

Partition the evaluation support into a natural-favored region $\mathcal{A}$ and a rewrite-favored region $\mathcal{B}$. On $\mathcal{A}$, approximate $p_s(x) = \rho_n p_n(x)$ with $\rho_n < 1$. On $\mathcal{B}$, approximate $p_n(x) = \rho_s p_s(x)$ with $\rho_s < 1$. Let λ_n and λ_s denote the evaluation mass of the two regions in the chosen logarithmic unit.

Equation 15 then becomes

$$p_R(x) = p_n(x)\left[1 - (1 - \rho_n)R\right], \qquad x \in \mathcal{A}, \tag{16}$$

$$p_R(x) = p_s(x)\left[\rho_s + (1 - \rho_s)R\right], \qquad x \in \mathcal{B}. \tag{17}$$

Substituting into the evaluation cross-entropy $-\mathbb{E}_{x\sim p_e} \log p_R(x)$ and absorbing terms independent of R into E_0 gives

$$E(R) = E_0 - \lambda_n \log\left[1 - (1 - \rho_n)R\right] - \lambda_s \log\left[\rho_s + (1 - \rho_s)R\right]. \tag{18}$$

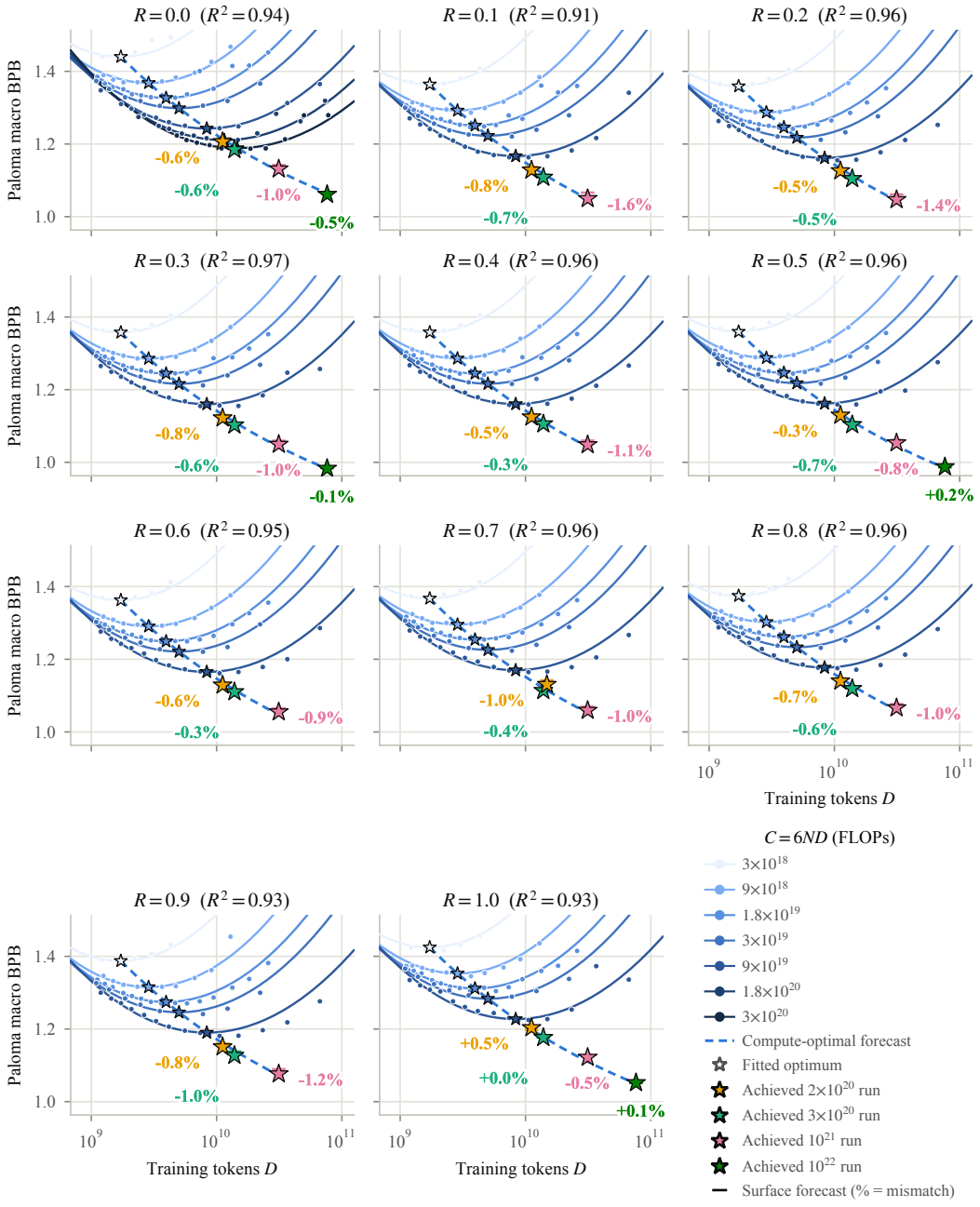

Figure 14: The shared $L(N, D, R)$ surface as per-R IsoFLOP slices at all eleven rewrite fractions, in Paloma macro BPB. Dots mark runs, open stars mark fitted frontier optima, and filled stars mark the 10^{21} and 10^{22} runs. Labels report measured minus fitted loss as a percentage of fitted loss.

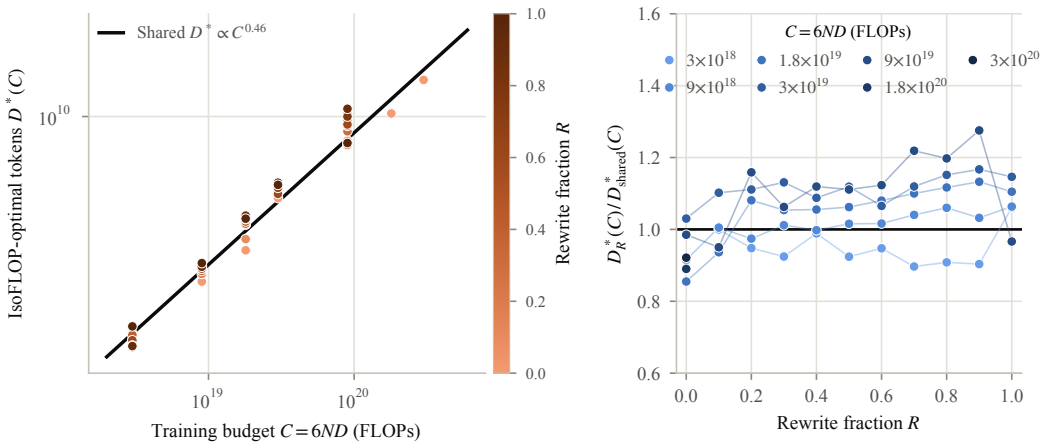

Figure 15: Per-R IsoFLOP optima against the shared frontier. **Left:** compute-optimal token count $D^*_R(C)$ from a separate Approach-2 parabola for each rewrite fraction and budget, colored by R, with the shared frontier $D^*(C) \propto C^{0.46}$. **Right:** ratio of each per-R optimum to the shared frontier, by budget.

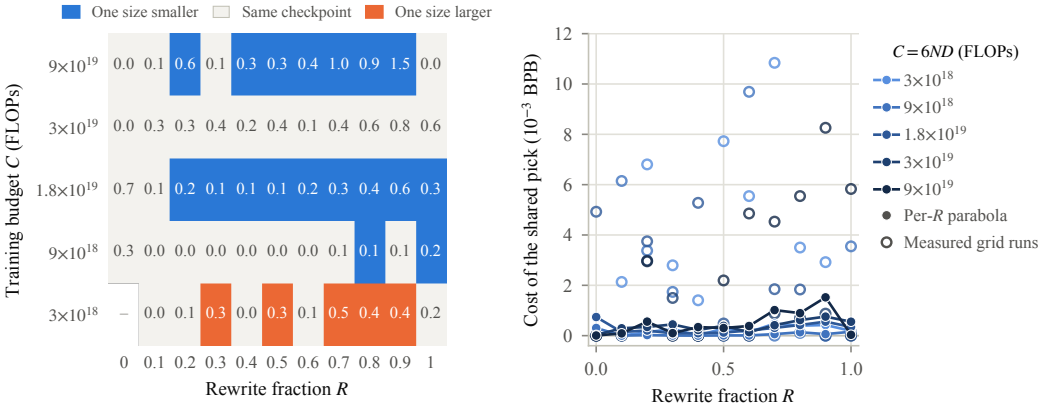

Figure 16: Whether fitting the frontier per R changes the selected model size. **Left:** for each sweep with a full rewrite ladder, the grid size nearest the per-R vertex relative to the one nearest the shared $N^*(C)$. Cells report the BPB cost of the shared choice on that sweep's own IsoFLOP parabola. **Right:** the same cost by rewrite fraction, on the parabola (filled) and between the two measured grid runs (open).

The derivatives are

$$E'(R) = \frac{\lambda_n(1-\rho_n)}{1-(1-\rho_n)R} - \frac{\lambda_s(1-\rho_s)}{\rho_s+(1-\rho_s)R}, \tag{19}$$

$$E''(R) = \frac{\lambda_n(1-\rho_n)^2}{[1-(1-\rho_n)R]^2} + \frac{\lambda_s(1-\rho_s)^2}{[\rho_s+(1-\rho_s)R]^2} > 0. \tag{20}$$

The floor is convex, with unconstrained minimizer

$$R^*_\infty = \frac{\lambda_s(1-\rho_s) - \lambda_n(1-\rho_n)\rho_s}{(1-\rho_n)(1-\rho_s)(\lambda_n+\lambda_s)}. \tag{21}$$

The constrained solution clips Eq. 21 to $[0, 1]$.

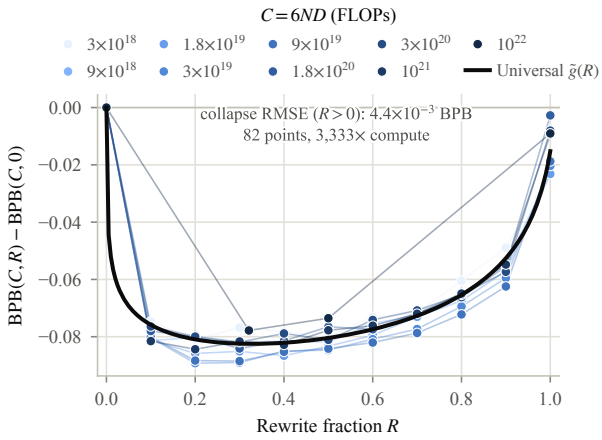

Figure 17: **Rewrite gains collapse onto one curve across 3,333× compute.** IsoFLOP-optimal Paloma BPB after subtracting each budget's $R = 0$ value. The fitted $\tilde{g}(R)$ has RMSE 0.0044 BPB over 82 observations from nine budgets.

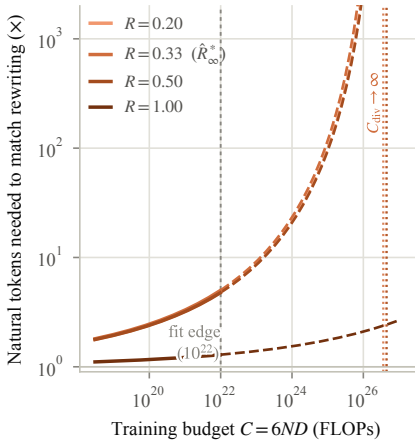

Figure 18: Natural tokens required to match the estimated floor-optimal mixture, as a function of training compute. The final fit gives 5.01× at 10^{22} FLOPs and diverges at $C_{\text{div}} = 3.61 \times 10^{26}$ FLOPs. Dashed curves exceed the fit range. The 671-run fit gives 12.17× and 1.75×10^{24} FLOPs.

N.1.3 Chinchilla Approach 2 → Approach 3 Conversion

Approach 2 fits the compute-optimal token and loss frontiers

$$D^*(C) = a_D C^p, \tag{22}$$

$$L^*(C) = E + a_L C^{-\gamma}. \tag{23}$$

With $C = 6ND$, the corresponding model frontier is

$$N^*(C) = \frac{1}{6a_D} C^{1-p}. \tag{24}$$

For an Approach-3 surface $L(N, D) = E + AN^{-\alpha} + BD^{-\beta}$, both reducible-loss terms share the frontier exponent. Matching exponents gives

$$\alpha = \frac{\gamma}{1-p}, \qquad \beta = \frac{\gamma}{p}. \tag{25}$$

The first-order condition gives the capacity and data terms shares $(1-p)$ and p of the frontier reducible loss, so

$$A = (1-p)a_L \left(\frac{1}{6a_D}\right)^{\alpha}, \tag{26}$$

$$B = p\, a_L a_D^{\beta}. \tag{27}$$

Table 5: Data-limited fits of Eq. (14) to final-checkpoint Paloma BPB, one fit per target–rewriter pair, 48 runs per row. RMSE, E_0, λ_n, λ_s, and P are in BPB; B is in BPB·tokens$^{\beta}$, with D in tokens as in Eq. (14); ρ_n, ρ_s, β, and s are dimensionless. ρ_s sits at its 10^{-4} lower bound except for the 1B pair.

Target	RMSE	E_0	λ_n	λ_s	ρ_n	ρ_s	B	β	P	s
550M	0.0147	1.259	0.0220	0.0045	0.0002	0.0001	1.55×10^{18}	1.939	0.0079	6.990
837M	0.0085	1.170	0.0458	0.0045	0.0105	0.0001	4.17×10^{8}	0.963	0.0096	1.539
1B	0.0115	1.209	0.0194	0.0000	0.0002	0.0249	1.86×10^{15}	1.636	0.0113	5.075
3B	0.0150	1.040	0.0823	0.0041	0.0872	0.0001	4.44×10^{5}	0.647	0.0174	2.082

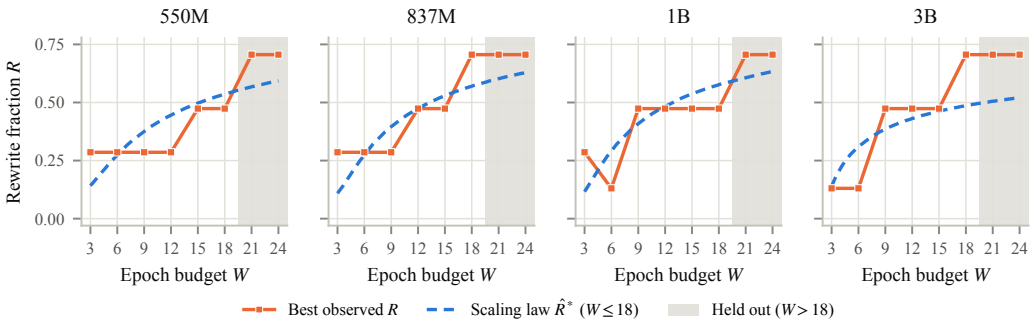

Figure 19: Estimated optimal rewrite fraction $\widehat{R}^*$ and best observed R as a function of epoch budget W for each target–rewriter pair. Orange squares mark the best observed R on the tested grid. Dashed blue curves show $\widehat{R}^*$ from each pair's scaling-law fit to $W \leq 18$, and the shaded budgets $W = 21$ and $W = 24$ are held out. $\widehat{R}^*$ increases with budget within every pair. Across pairs, the rewriter changes with the target size.

For the final FineWeb-Edu fit, $a_D = 4.5033$, $p = 0.46447$, $a_L = 67.2942$, and $\gamma = 0.10963$, giving $\alpha = 0.20472$ and $\beta = 0.23604$.

N.1.4 Fitting $L(N, D, R, T)$

Data-limited case. Each target–rewriter pair contributes 48 runs: eight epoch budgets by six rewrite fractions. We fit Eq. (14) to final-checkpoint Paloma BPB separately for each pair, because the size-matched rewriter changes the quality of the synthetic data with the target size (Figure 8). Each fit pools all epoch budgets and rewrite fractions and uses nine parameters: the five floor parameters, B, β, P, and s (Table 5). We minimize the unweighted sum of squared residuals with bounded least squares ($\rho_n, \rho_s \in [10^{-4}, 1]$, $\beta \in [0.05, 3]$, $s \in [0, 8]$) from six fixed starts and 60 seeded random starts, keeping the lowest-cost solution. Table 5 reports point estimates only; the fit script and its cached outputs are released with the code.

Paloma BPB is U-shaped in rewrite fraction. The fitted optimum increases with epoch budget within every target–rewriter pair. The repetition coefficient P increases across pairs. The best-mixture asymptote is no higher than the natural-only asymptote at every pair and lies strictly below it at three; the 1B fit places the two at parity and attributes its rising $\widehat{R}^*(W)$ to the repetition penalty. The fit does not isolate a target-size effect, because the size-matched rewriter changes the quality of the synthetic data with the target size.

Out of sample, we refit Eq. (14) for each target–rewriter pair to the 36 runs at $W \leq 18$ and forecast the six rewrite fractions at each of the held-out budgets $W = 21$ and $W = 24$. Figure 4 reports the R this fit selects at every budget and its measured cost relative to the best observed R. The mean error is within $\pm 1.4\%$ at every pair and budget, and the measured loss at the best observed R, which is $R = 0.71$ at every pair and both budgets, is within 0.74% of its forecast (Table 6). The largest

Table 6: Forward data-limited forecasts. Each row fits the 36 runs at $W \leq 18$ and forecasts the six rewrite fractions at the held-out budget W. RMSE is in BPB.

Target	Held-out W	Fit RMSE	Held-out RMSE	Mean error	Per-R range	Error at best obs. R
550M	21	0.0114	0.0267	-0.73%	$[-3.98, +1.98]\%$	-0.28%
550M	24	0.0114	0.0294	-1.39%	$[-3.90, +0.85]\%$	-0.26%
837M	21	0.0083	0.0060	$+0.13\%$	$[-0.29, +0.86]\%$	-0.27%
837M	24	0.0083	0.0140	-0.42%	$[-1.98, +0.79]\%$	-0.68%
1B	21	0.0102	0.0225	-0.50%	$[-3.01, +2.05]\%$	$+0.05\%$
1B	24	0.0102	0.0212	-0.81%	$[-2.97, +0.91]\%$	$+0.03\%$
3B	21	0.0143	0.0163	$+0.27\%$	$[-1.94, +2.14]\%$	-0.50%
3B	24	0.0143	0.0201	-0.04%	$[-2.22, +2.65]\%$	-0.74%

Table 7: Final FineWeb-Edu surface parameters. N and D are absolute parameter and token counts, and $C = 6ND$ is in FLOPs. The fitted ρ_s reaches its lower bound. The fitted floor at $R = 0$ is $E(0) = E_0 - \lambda_s \log \rho_s = 0.805691$ (Eq. 10).

Parameter	Value	Parameter	Value
E_0	0.701293	A	18.3516
λ_n	0.024066	α	0.204717
ρ_n	0.024598	B	44.5853
λ_s	0.011335	β	0.236040
ρ_s	10^{-4}	H	3.00277
		η	0.072262

errors are at $R = 0.13$ for the 550M, 837M, and 1B pairs and at $R = 0.29$ for the 3B pair. The error is $(L_{\text{measured}} - L_{\text{predicted}})/L_{\text{predicted}}$, as in Table 8.

Compute-limited case. The surface fits the 604 observations in three stages: the mixture-KL floor, the pooled Approach-2 frontiers promoted through Appendix N.1.3, and the curvature term of Appendix N.1.6. Table 7 lists the parameters. The RMSE over all runs is 0.01449 BPB. The promoted frontiers are

$$D^*(C) = 4.5033\, C^{0.46447}, \tag{28}$$

$$L^*(C, 0) = 0.80569 + 67.2942\, C^{-0.10963}, \tag{29}$$

$$N^*(C) = \frac{C}{6D^*(C)}, \tag{30}$$

which give $(N^*, D^*) = (0.554\text{B}, 3.009\text{B})$ at 10^{19} FLOPs and $(22.39\text{B}, 74.44\text{B})$ at 10^{22} FLOPs.

Surface fit. We fit the joint surface $L(N, D, R)$ to the 604 runs of the IsoFLOP ladders, which include the 10^{22}-FLOP hero run as the largest-budget point. In sample, the fit has RMSE 0.0145 BPB ($R^2 = 0.95$) over all 604 runs, a root-mean-square relative error of 1.1%. The residual concentrates in the over-trained tail: on the 296 runs within $2\times$ of the fitted token optimum $D^*(C)$, the RMSE is 0.0096 BPB ($R^2 = 0.98$) and the relative error 0.7%. The quantities we use downstream are the compute-optimal losses, the lowest measured loss at each of the 58 pairs of budget and rewrite fraction. The fitted frontier reproduces these to within 0.75% (median 0.19%). Stated in compute rather than loss, the largest of these errors is the amount by which the fitted compute-optimal loss changes when the training FLOPs change by 19%, and the median error is the amount from a 4% change in training FLOPs. Figure 14 shows the fitted ladders at all eleven rewrite fractions. Out of sample, we perform a forward leave-one-budget-out analysis: we fit the surface to the budgets below each held-out budget and forecast every R. These forecasts have mean errors within $\pm 2\%$ at 2–3$\times$ extrapolation, and forecasting the 10^{22} runs from a fit range ending at 3×10^{20} FLOPs gives a mean error of -1.0%, with every measured loss 0.9–1.4% below its forecast (Table 8).

Table 8: Forward compute-limited forecasts by training budget. The 10^{22} row fits through 3×10^{20} FLOPs and predicts the four rewrite fractions trained at the hero-run allocation ($R \in \{0, 0.321, 0.5, 1\}$).

Test training FLOPs	Largest fit training FLOPs	Extrap.	Mean error	Per-R range
3×10^{19}	1.8×10^{19}	$1.7\times$	-0.50%	$[-0.81, -0.16]\%$
9×10^{19}	3×10^{19}	$3.0\times$	-0.36%	$[-0.77, +0.02]\%$
1.8×10^{20}	9×10^{19}	$2.0\times$	-0.21%	$[-0.56, +0.80]\%$
3×10^{20}	1.8×10^{20}	$1.7\times$	-0.70%	$[-1.18, -0.13]\%$
10^{21}	3×10^{20}	$3.3\times$	-1.32%	$[-1.87, -0.73]\%$
10^{22}	3×10^{20}	$33.3\times$	-1.04%	$[-1.40, -0.85]\%$

Table 8 reports forward leave-one-budget-out forecasts. Each row fits only the budgets below the held-out budget and forecasts every rewrite fraction at that budget. The error is $(L_{\text{measured}} - L_{\text{predicted}})/L_{\text{predicted}}$, so negative values are conservative forecasts.

N.1.5 Repetition-Penalty Form Ablation

Eq. (14) charges repeated natural tokens with the additive penalty $F = P\,(1-R)^s((1-R)D/T - 1)_+$, with $(1-R)D/T = 3W(1-R)/(3+2R)$ natural epochs. The factor $((1-R)D/T-1)_+$ is the one-parameter penalty of Lovelace et al. (2026) (their Eq. 6), linear in the epochs beyond the first, with the capacity ratio N/T absorbed into P because both are fixed within a data-limited fit; the dilution factor $(1-R)^s$ is ours. We refit the 48 runs of every target–rewriter pair, with the floor $E(R)$, B, and β free and the protocol of Table 5, under six penalties: (i) the paper's $P(1-R)^s((1-R)D/T-1)_+$; (ii) $P(1-R)^s((1-R)D/T-1)_+^\delta$ with $\delta \in [0.5, 3]$ free, the superlinear exponent on the repetition count of their Eq. 8; (iii) the same with δ fixed at 1.84, the value that Lovelace et al. (2026) fit for their Eq. 8 on their own runs; we do not fit it, and this variant tests whether their exponent transfers to our setting (labeled "$\delta = 1.84$ (Lovelace)" in Figures 20–22); (iv) $P((1-R)D/T-1)_+$, their Eq. 6 without dilution ($s = 0$); (v) no penalty, $P = 0$, so that D enters the Chinchilla term only; and (vi) no additive penalty but the effective-data term of Muennighoff et al. (2023) in place of D, with the saturating count applied to the natural epochs only, since rewrites are never repeated. Their Eq. 7 exponent on N/U_D is not identifiable here, because N and T do not vary within a pair. Per pair, 8 runs have $(1-R)D/T = 0$ ($R = 1$), one has $0 < (1-R)D/T < 1$ ($W = 3$, $R = 0.71$), and 39 have $(1-R)D/T > 1$ (up to $(1-R)D/T = 24$ at $R = 0$), so every penalty is identified on the same 39 cells and the data cannot separate $((1-R)D/T-1)_+$ from $(1-R)D/T$ (Section N.1.4).

Three conclusions follow (Table 9, Figures 20–22). Figure 22 shows the fits run by run: the paper form and the two δ variants have indistinguishable R^2 (0.959–0.985 across pairs for the paper form, within 0.011 of it for δ free or fixed), the no-dilution form loses 0.006–0.023, and the two forms without an additive penalty fall to $R^2 = 0.62$–0.87, lowest at 3B, where the penalty is largest. First, an additive penalty is needed: dropping it raises RMSE by 1.8–3.4×, pushes β to its upper bound of 3, and under-predicts the held-out $W = 24$ runs by 0.8–3.8% on average because the $R = 0$ run at $W = 24$ cannot be reached by a term in D alone; the effective-data variant fits no better, since its saturation parameter goes to 0.85–0.92 with β at its lower bound at three pairs and to its upper bound of 200, which is the no-penalty form, at the fourth. Second, the exponent on the repetition count is not identified by this grid. Freeing δ lowers RMSE by at most 0.0005 BPB and returns $\delta = 0.77, 1.10, 1.05, 1.19$ across pairs, indistinguishable from the linear form on the tested fractions, while s moves to compensate ($s = 8$, its bound, at 550M). Fixing $\delta = 1.84$, the value Lovelace et al. (2026) obtain from residuals against a single-epoch Chinchilla fit with N/U_D varied, is worse at every pair (+0.0015 to +0.0023 BPB), drives β to its bound at 1B, and doubles the held-out mean error; their exponent does not transfer once the floor and s are free. Third, the dilution factor $(1-R)^s$ is what the linear form needs to coexist with the mixture floor: without it ($s = 0$) RMSE rises by 0.0011–0.0047 BPB and the fit is degenerate, with $E_0 \to 0$ and β at its lower bound at two pairs, so the Chinchilla term becomes a near-constant and the floor absorbs the penalty's R-dependence. On the selection metric of Figure 4 the picture is less flattering to the paper's form: with the $W \le 18$ fit it selects $R = 0.47$ instead of the best observed $R = 0.71$ at

Table 9: Repetition-penalty form ablation for Eq. (14), one fit per target–rewriter pair, 48 runs per fit. RMSE is in BPB and $R^2 = 1 - \mathrm{SS}_{\mathrm{res}}/\mathrm{SS}_{\mathrm{tot}}$ over the 48 runs. The held-out columns refit the 36 runs at $W \le 18$ and forecast the six rewrite fractions at $W = 21$ and $W = 24$: mean error is the mean of $(L_{\mathrm{measured}} - L_{\mathrm{predicted}})/L_{\mathrm{predicted}}$ over the six, error at best obs. R is the same quantity at the best observed R ($R = 0.71$ at every pair and both budgets), and R_{sel} is the tested fraction with the lowest forecast loss, the selection rule of Figure 4. $\widehat{R}^*_{24}$ is the minimizer of the full 48-run fit over $R \in [0, 1]$ at $W = 24$. The parameter count includes the five floor parameters, B, and β. Fitted parameters are in `repetition_penalty/de_penalty_ablation.csv`.

Penalty	Params	Target	RMSE	R^2	Mean error		Error at best obs. R		R_{sel}		$\hat{R}^*_{24}$
					$W = 21$	$W = 24$	$W = 21$	$W = 24$	$W = 21$	$W = 24$	
$P(1-R)^s((1-R)D/T-1)_+$ (paper)	9	550M	0.0147	0.959	−0.73%	−1.39%	−0.28%	−0.26%	0.47	0.71	0.41
		837M	0.0085	0.985	+0.13%	−0.42%	−0.27%	−0.68%	0.71	0.71	0.59
		1B	0.0115	0.974	−0.50%	−0.81%	+0.05%	+0.03%	0.71	0.71	0.49
		3B	0.0150	0.968	+0.27%	−0.04%	−0.50%	−0.74%	0.47	0.47	0.57
$P(1-R)^s((1-R)D/T-1)_+^\delta$ δ free	10	550M	0.0145	0.960	−0.73%	−1.39%	−0.28%	−0.26%	0.47	0.71	0.39
		837M	0.0084	0.985	+0.09%	−0.48%	−0.27%	−0.68%	0.71	0.71	0.59
		1B	0.0115	0.974	−0.24%	−0.40%	+0.01%	−0.01%	0.71	0.71	0.49
		3B	0.0145	0.970	−0.30%	−0.98%	−0.34%	−0.56%	0.47	0.47	0.56
$P(1-R)^s((1-R)D/T-1)_+^{1.84}$	9	550M	0.0164	0.949	−1.45%	−2.60%	−0.67%	−0.75%	0.47	0.71	0.44
		837M	0.0100	0.979	−0.37%	−1.24%	−0.72%	−1.16%	0.47	0.47	0.59
		1B	0.0132	0.966	−1.09%	−1.64%	−1.14%	−1.32%	0.47	0.47	0.51
		3B	0.0173	0.958	−1.01%	−2.19%	−0.52%	−0.80%	0.47	0.47	0.59
$P((1-R)D/T-1)_+$ (no dilution)	8	550M	0.0158	0.953	−0.84%	−1.61%	−0.34%	−0.44%	0.71	0.71	0.62
		837M	0.0108	0.975	−0.01%	−0.67%	−0.37%	−0.92%	0.71	0.71	0.70
		1B	0.0154	0.954	−0.69%	−1.13%	−0.16%	−0.37%	0.71	0.71	0.69
		3B	0.0197	0.945	−0.54%	−1.38%	−0.21%	−0.75%	0.71	0.71	0.74
No penalty	7	550M	0.0265	0.867	+0.86%	+0.83%	−1.24%	−1.31%	0.47	0.47	0.44
		837M	0.0287	0.824	+1.93%	+2.13%	−1.09%	−1.54%	0.47	0.47	0.53
		1B	0.0341	0.773	+1.45%	+1.96%	−1.26%	−1.45%	0.47	0.47	0.52
		3B	0.0516	0.622	+3.01%	+3.75%	−1.60%	−1.99%	0.47	0.47	0.56
Effective data (Muennighoff)	8	550M	0.0261	0.871	+1.66%	+1.85%	−0.07%	+0.14%	0.47	0.47	0.46
		837M	0.0288	0.823	+2.76%	+3.17%	+0.04%	−0.17%	0.47	0.47	0.53
		1B	0.0336	0.779	+2.50%	+3.31%	+0.28%	+0.45%	0.47	0.47	0.53
		3B	0.0514	0.626	+4.36%	+5.44%	+0.32%	+0.36%	0.47	0.47	0.57

3B (measured cost 0.50% at $W = 21$ and 1.14% at $W = 24$) and at 550M, $W = 21$ (0.01%), as do the δ variants, whereas the no-dilution form selects 0.71 at all eight held-out cells despite its worse RMSE, because its $\widehat{R}^*(W)$ rises faster. All forms with dilution place the continuous $\widehat{R}^*(24)$ at 0.39–0.59, below the best observed R, and the two forms without a penalty are flat in W. We keep the linear form of Eq. (14): it is the most parsimonious form that reproduces the U-curves, and the extra exponent buys nothing the data can resolve. The exponent s, not δ, governs how fast $\widehat{R}^*$ grows with budget, and it is weakly identified ($s = 1.5$–7.0 across pairs), so extrapolations of $\widehat{R}^*(W)$ beyond $W = 24$ should be read with that in mind.

N.1.6 Ablating the Curvature Term

The compute-limited surface adds to Eq. (1) with $F = 0$ a term fit to the excess curvature of the measured IsoFLOP parabolas,

$$\mathrm{Curv}(N, D) = \frac{H}{C^\eta}\left[\log_{10}\frac{D}{D^*(C)}\right]^2, \qquad C = 6ND, \tag{31}$$

so that

$$L_{\mathrm{CE}}(N, D, R) = E(R) + \frac{A}{N^\alpha} + \frac{B}{D^\beta} + \mathrm{Curv}(N, D). \tag{32}$$

The term vanishes with zero slope on the frontier, so it changes off-frontier predictions without moving $N^*(C)$, $D^*(C)$, or $L^*_{\mathrm{base}}(C)$.

We ablate the term on the 671-run fit of Appendix N.1.1, holding the floor and Chinchilla terms fixed. Removing it raises the residual sum of squares from 0.0608 to 0.2726, with likelihood-ratio

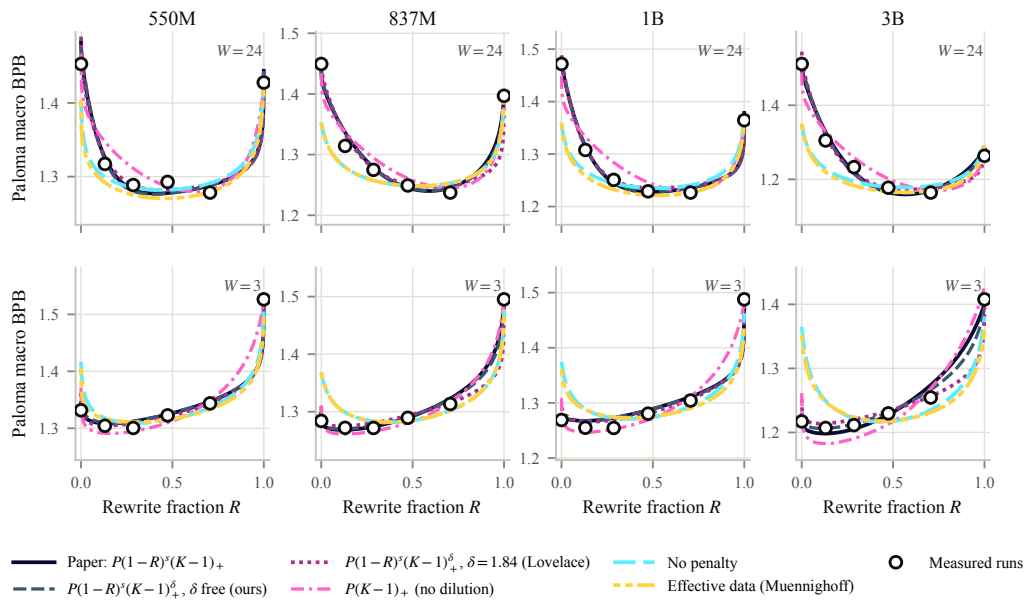

Figure 20: Fitted Paloma BPB against rewrite fraction under each repetition penalty (full 48-run fits), at $W = 24$ (top) and $W = 3$ (bottom), with the measured runs. The four penalties with an additive term are nearly coincident at $W = 24$ between the tested fractions; the two forms without one fall 0.05–0.16 BPB below the $R = 0$ run at $W = 24$ and 0.07–0.15 BPB above it at $W = 3$. The no-dilution form places its $W = 3$ minimum below the measured points at small R.

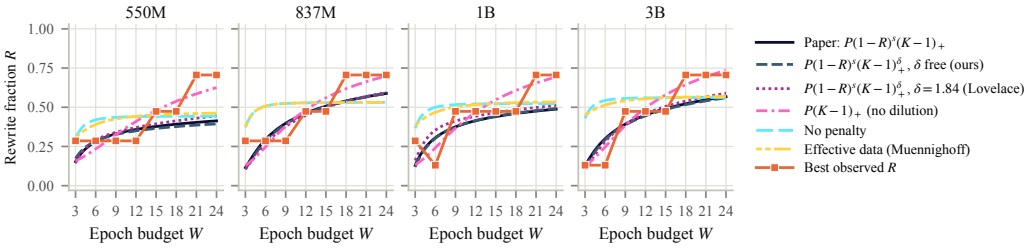

Figure 21: Estimated optimal rewrite fraction $\widehat{R}^*(W)$ under each repetition penalty (full 48-run fits), with the best observed R at each budget. All forms with the dilution factor $(1 - R)^s$ give $\widehat{R}^*(24)$ between 0.39 and 0.59, below the best observed $R = 0.71$ at $W \geq 18$; the no-dilution form rises faster (0.62–0.74) but at the cost of fit quality and a degenerate floor (Table 9). The forms without an additive penalty are flat in W.

statistic 1006.6. The architecture-wise wild bootstrap of Appendix M, over 79 architectures and 400 draws, rejects the restriction ($p = 1/401$).

N.2 Domain-Specific Compute Optima

Break-even with SmolLM2-Instruct by evaluation. On QA, the lowest fitted QA BPB over model size, token count, and rewrite fraction matches SmolLM2-1.7B-Instruct at 5.3×10^{21} total FLOPs with $\widehat{R}^* = 0.74$, and the 10^{22}-FLOP run trained at the QA optimum measures 0.696 QA BPB against 0.709 for SmolLM2-Instruct. On Math and Code the fitted minimum matches SmolLM2-Instruct only at $\approx 6 \times 10^{22}$ and $\approx 4 \times 10^{23}$ total FLOPs, $6\times$ and $43\times$ the largest budget we trained, and the Code crossing lies above SmolLM2's $\approx 1.1 \times 10^{23}$ training FLOPs. SmolLM2's

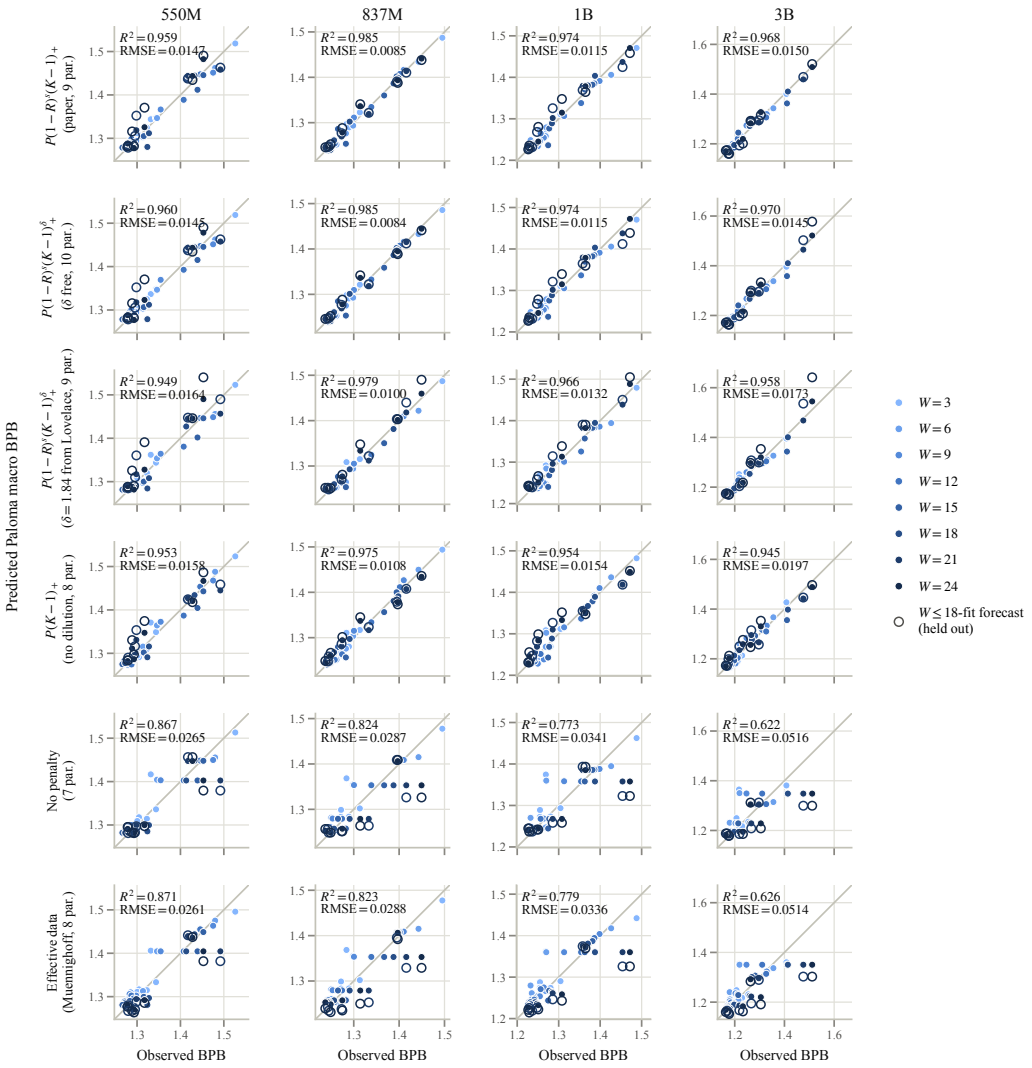

Figure 22: Observed against predicted Paloma macro BPB for every repetition penalty (rows, in the order of Table 9) and target–rewriter pair (columns), all 48 runs of the full fit, coloured by epoch budget W, with the identity line. Hollow markers are the $W = 21$ and $W = 24$ forecasts of the $W \leq 18$ refit. Each panel reports R^2 and RMSE of the full fit; axis limits are shared within a column. The three dilution forms and the no-dilution form lie along the identity line; the two forms without an additive penalty flatten to horizontal bands at large W, where the $R = 0$ runs cannot be separated from their neighbours by a term in D alone.

training corpus includes FineMath and Stack-Edu (Ben Allal et al., 2025) and StarCoderData (Li et al., 2023), math and code data that FineWeb-Edu largely lacks. For each domain, the 10^{22}-FLOP run trained at that domain's fitted optimum measures a higher loss than its surface forecasts, by 4% on QA, 14% on Math, and 32% on Code, against 1% for the Paloma-optimal run. We hypothesize two causes. First, we tuned the training recipe on Paloma, whose macro average weights 16 sources equally: ten web or encyclopedic text sources, five social-media sources, one code source, and no math source (Appendix D). Web and encyclopedic text carries the knowledge that QA evaluations test, so the forecast error is smallest on Paloma and then QA. Second, the rewrite styles include math and not code. The rewrites contain at most 0.8% code lines and add lists, tables, and headers rather than code (Appendix K), so code remains rare in the training data even at $R = 1$, and the Code surface is fit to the narrowest range of loss.

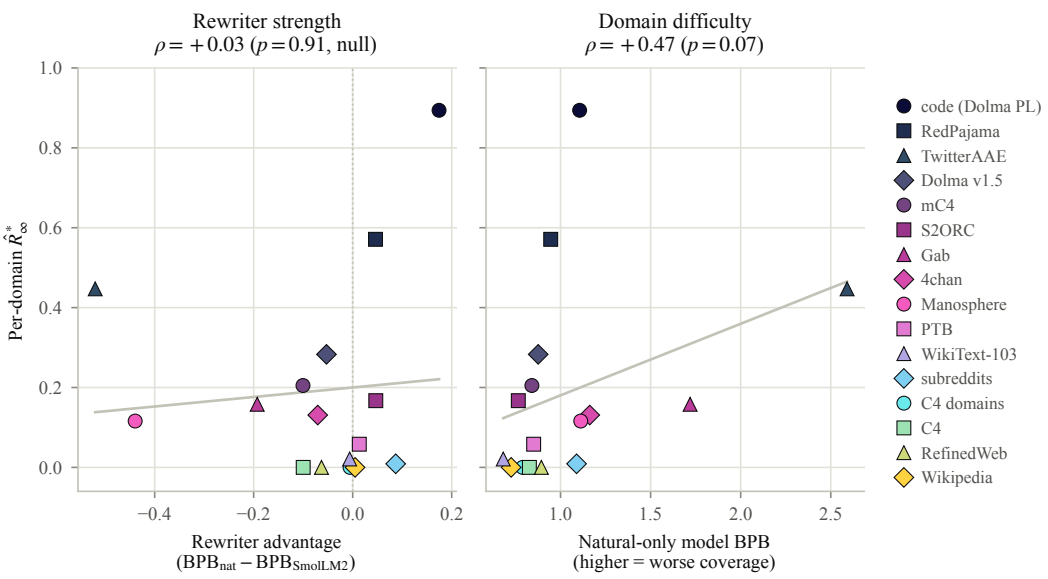

Figure 23: The estimated floor-optimal rewrite fraction $\widehat{R}^*_\infty$ against two properties of each Paloma source. **Left:** $\widehat{R}^*_\infty$ has no association with the rewriter's relative strength on the source (Spearman $\rho = 0.03$, $p = 0.91$). **Right:** $\widehat{R}^*_\infty$ increases with the natural-data model's BPB on the source (Spearman $\rho = 0.47$, $p = 0.07$).

Figures 24 and 25 decompose the aggregate trajectories of Figure 2 into Paloma sources and domain-suite tasks. Across Paloma sources, $\widehat{R}^*_\infty$ reaches $\approx$0.89 for programming languages and 0.57 for RedPajama, while C4, RefinedWeb, and Wikipedia remain at $\widehat{R}^* = 0$ throughout the evaluated compute range (Figure 24). Every Math and Code task reaches $\widehat{R}^*_\infty = 1$. Within QA, ARC-Easy, ARC-Challenge, and MMLU reach $\widehat{R}^*_\infty \geq 0.82$, HellaSwag remains below 0.1, and the QA aggregate reaches 0.77 (Figure 25).

Each trajectory comes from a separate surface fit to its aggregate. The Paloma, QA, Math, and Code fits contain 604, 597, 608, and 610 observations and have RMSE 0.0148, 0.0195, 0.0301, and 0.0681 BPB. Their estimated floor optima are 0.329, 0.768, 1.0, and 1.0. The Math and Code estimates lie at the boundary, and their break-even points lie beyond the measured ladders.

Compute-optimal allocation by evaluation. Each per-evaluation surface fit has its own compute-optimal frontier $N^*(C)$ and $D^*(C)$, shared across rewrite fractions within the fit. Figure 26 shows each evaluation's natural-only IsoFLOP slices with its fitted optima and the Paloma frontier overlaid, and Table 10 reports the frontier at two training budgets. The QA frontier allocates the fewest tokens per parameter and the Code frontier the most, with Math between them: at 10^{22} FLOPs the QA optimum has $1.5\times$ the parameters of the Code optimum at the same training compute. This reproduces the finding of Roberts et al. (2025) that knowledge QA is capacity-hungry while code is data-hungry, and that the compute-optimal parameter count depends on the skill composition of the validation set by up to $\approx$50%. Our fits reach the same conclusion on ladders that include rewrites at every fraction, so the allocation difference is a property of the evaluation rather than of the rewrite fraction. Relative to the Paloma frontier, which increases more steeply in N, the QA optimum lies at more parameters below $\approx 10^{20}$ FLOPs and fewer above, while the Code optimum lies at fewer parameters at every budget.

N.3 Evaluations of the 10^{22}-FLOP Runs

Table 11 compares the 10^{22}-FLOP runs with SmolLM2 using the matched Paloma reference and the offline QA, Math, and Code references. Table 12 reports the 10^{22} runs by Paloma source. Tables 13 and 14 break the QA, Math, and Code aggregates into their tasks, subjects, and languages.

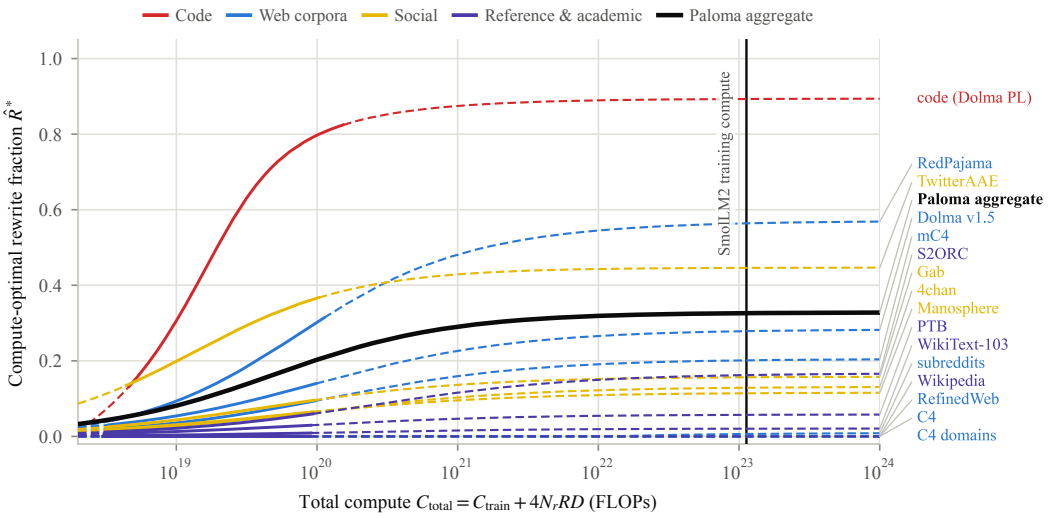

Figure 24: Estimated compute-optimal rewrite fraction $\widehat{R}^*(C_{\text{total}})$ for each Paloma source. Solid curves cover the measured ladders through $\approx 2 \times 10^{20}$ total FLOPs, and dashed curves extrapolate. The thick curve is Paloma macro BPB. The vertical line marks SmolLM2's training compute.

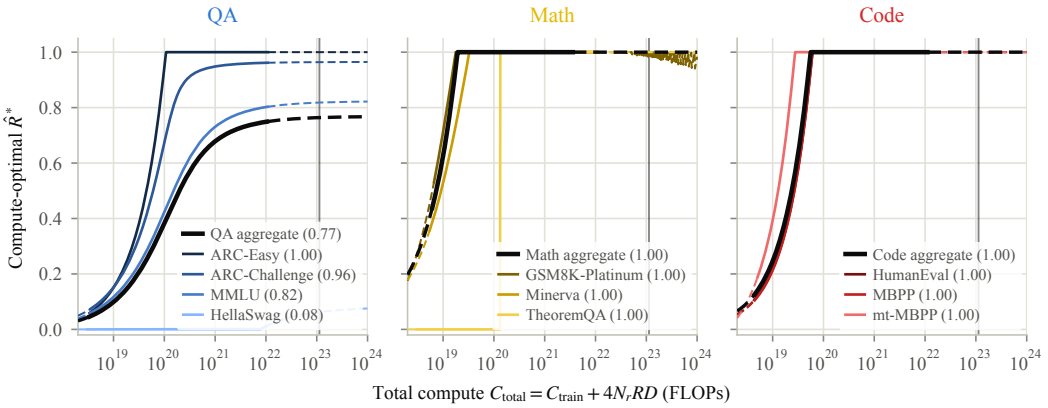

Figure 25: Estimated compute-optimal rewrite fraction $\widehat{R}^*(C_{\text{total}})$ for each task. Legends report $\widehat{R}^*_\infty$, and thick curves show the aggregate trajectories of Figure 2. MMLU, Minerva, and multilingual MBPP average 57, 7, and 17 subtasks. TheoremQA switches from $R = 0$ to $R = 1$ because its fitted floor is nearly linear in R; its fit excludes the two budgets with the fewest model widths.

N.4 Breakeven Parameters

N.4.1 Definition and Relation to Training Compute

For each comparison metric m, break-even is the smallest total compute, accounting for rewrites, at which the fitted compute-optimal frontier reaches SmolLM2-Instruct's score:

$$C_{\text{BE},m} = \min\{C_{\text{total}} : L^*_m(C_{\text{total}}) \le L_{m,\text{SmolLM2-inst}}\}. \tag{33}$$

Here $L^*_m(C_{\text{total}})$ is the loss at the joint optimum of Eq. (13) on metric m's fitted surface. SmolLM2-Instruct enters only as the reference score. Its own training compute is 1.1286×10^{23} FLOPs. The crossing accounts for rewrite generation but not SmolLM2's training. Paloma uses the 0.9716 BPB reference matched to our byte-pooled aggregation and 4096-token context. QA, Math, and Code use the same offline evaluations for both model families.

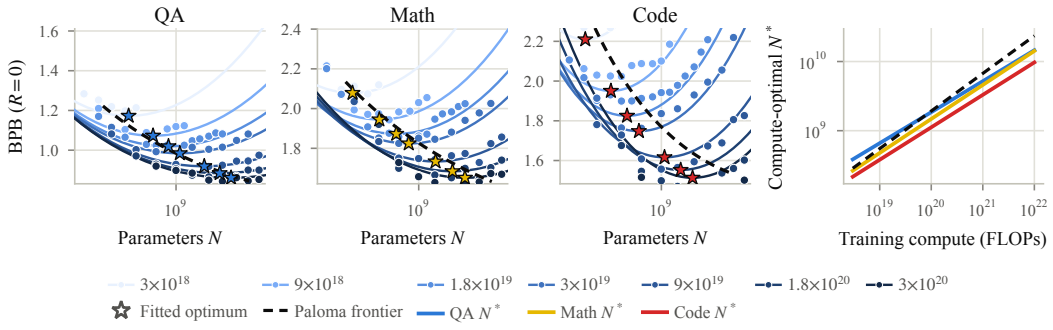

Figure 26: **Knowledge QA is capacity-hungry and Code is data-hungry. Left three panels:** natural-only ($R = 0$) IsoFLOP slices from each evaluation's own surface fit, loss against parameter count, colored by training budget, with dots for runs and stars for the fitted optimum at each budget. The dashed line is the Paloma frontier $N^*(C)$ drawn on that evaluation's surface. **Right:** compute-optimal parameter count of each evaluation's fit against training compute. The QA optimum has more parameters than the Code optimum at every budget, $1.5\times$ at 10^{22} FLOPs.

Table 10: Compute-optimal allocation from each evaluation's surface fit at two training budgets. D^*/N^* is tokens per parameter.

Evaluation	$C = 10^{19}$			$C = 10^{22}$		
	N^*	D^*	D^*/N^*	N^*	D^*	D^*/N^*
Paloma	0.55B	3.0B	5.5	23.5B	70.9B	3.0
QA	0.66B	2.5B	3.8	14.6B	114.4B	7.8
Math	0.48B	3.5B	7.3	14.2B	117.2B	8.2
Code	0.38B	4.4B	11.5	9.7B	171.7B	17.7

N.4.2 Breakeven N, D, and R per Domain

Table 15 gives the fitted allocation at each matched crossing. The training-compute column is $6N^*D^*$, and its difference from total compute is the generation cost. Paloma, QA, and Math cross below SmolLM2's training compute, and Code crosses above it.

N.5 Loss- vs Accuracy-Based Fits

N.5.1 Parametric Accuracy Model

As a check on the BPB criterion, we fit QA accuracy directly on all 607 accuracy observations by mapping the surface through a logistic link:

$$\widehat{a}(N, D, R) = \frac{\theta}{1 + \exp z(N, D, R)}, \tag{34}$$

$$z(N, D, R) = E(R) + \frac{A}{N^\alpha} + \frac{B}{D^\beta} + HC^{-\eta}\left[\log_{10}\frac{D}{D^*(C)}\right]^2, \tag{35}$$

where $C = 6ND$ and $E(R)$ is the mixture-dependent floor of Eq. (10). The fit minimizes a soft-L_1 objective with scale 0.02, initialized from the QA-BPB surface, and refits every parameter.

N.5.2 Accuracy Fits

The observed QA accuracies span 0.3041–0.5129. The fit has $R^2 = 0.98795$, RMSE 0.00453, and mean absolute error 0.00345. Its ceiling θ reaches the upper bound 1 and its intercept E_0 the lower bound -10, so the asymptotic parameters are weakly identified although the in-grid fit is accurate.

Table 11: **The 10^{22}-FLOP hero run nearly matches SmolLM2-Instruct on Paloma.** The Paloma column uses byte-pooled BPB at a 4096-token context for both model families. The QA, Math, and Code columns retain their matched offline evaluations. The first four rows share the Paloma-optimal (N, D), hence training compute $C \approx 10^{22}$ FLOPs, and differ only in R; their total compute $C_{\text{total}} = 6ND + 4N_rRD$ varies with R. SmolLM2 is trained on 11T tokens at $\approx 1.1 \times 10^{23}$ FLOPs, $11\times$ the total budget of the hero run.

Optimized for	Model	R	N	D	BPB ↓				Acc. ↑
					Paloma	QA	Math	Code	QA
–	Natural-only	0	21.9B	76B	1.061	0.731	1.328	1.042	0.619
Paloma	Hero	0.32	21.9B	76B	0.983	0.700	0.842	0.587	**0.622**
–	Grid	0.5	21.9B	76B	0.988	0.698	0.820	0.568	0.620
–	Grid	1.0	21.9B	76B	1.052	0.719	0.800	0.529	0.594
QA	Grid	0.74	13.8B	120B	0.988	0.696	0.801	0.520	0.620
Code	Grid	1.0	9.7B	172B	1.107	0.710	0.788	0.507	0.597
–	SmolLM2-1.7B	–	1.7B	11T	**0.943**	**0.667**	0.707	**0.399**	0.617
–	SmolLM2-1.7B-Instruct	–	1.7B	11T	0.972	0.709	**0.697**	0.426	0.606

Table 12: **The $R = 0.321$ hero run minimizes Paloma macro BPB among the runs at 10^{22} training FLOPs.** The four 22B runs share $(N, D) = (21.9\text{B}, 76\text{B})$; the best value among them per row is bold. The SmolLM2 columns are evaluated on the same Paloma validation split at a 4096-token context. The $R = 0.321$ run reaches a lower BPB than SmolLM2-1.7B-Instruct on 10 of the 16 sources.

Domain	10^{22} training FLOPs, 22B				SmolLM2-1.7B	
	$R = 0$	$R = 0.321$	$R = 0.5$	$R = 1.0$	Base	Instruct
4chan	1.162	**1.023**	1.034	1.143	0.926	0.952
C4 100 Domains	**0.790**	0.793	0.797	0.844	0.772	0.797
C4 English	**0.827**	0.828	0.833	0.874	0.817	0.844
Dolma v1.5	0.876	**0.804**	0.808	0.863	0.763	0.791
Dolma 100 PLs	1.106	0.694	**0.680**	0.698	0.486	0.511
Dolma 100 Subreddits	1.089	**1.022**	1.026	1.066	0.973	1.001
Falcon RefinedWeb	0.894	**0.887**	0.891	0.943	0.869	0.897
Gab	1.719	**1.579**	1.590	1.658	1.550	1.587
M2D2 S2ORC	0.766	**0.760**	0.763	0.823	0.737	0.764
M2D2 Wikipedia	0.726	**0.716**	0.723	0.789	0.728	0.758
Manosphere	1.112	**1.057**	1.063	1.123	1.000	1.030
mC4	0.841	**0.825**	0.829	0.890	0.801	0.828
Penn Treebank	0.851	**0.843**	0.848	0.955	0.851	0.882
RedPajama	0.946	**0.775**	0.777	0.826	0.710	0.737
TwitterAAE	2.591	**2.453**	2.459	2.580	2.426	2.455
WikiText-103	0.682	**0.673**	0.679	0.759	0.681	0.711
Macro average	1.061	**0.983**	0.988	1.052	0.943	0.972

With the 1.71B rewriter cost, the accuracy fit gives $\widehat{R}^*(10^{22}) = 0.638$. It reaches the SmolLM2-instruct and base QA accuracies, 0.606 and 0.617, at 3.336×10^{21} and 4.484×10^{21} FLOPs. Both references exceed every accuracy in the fitting grid, so both crossings are extrapolations. The QA hero run obtains 0.62017 accuracy against a forecast of 0.64734. The QA-BPB surface gives $\widehat{R}^*(10^{22}) = 0.749$ and crosses the instruct BPB reference at 5.316×10^{21} FLOPs. Both criteria favor a large rewrite fraction, and the exact estimated optimum and crossing depend on the criterion.

Table 13: **Task-level scores behind the QA, Math, and Code aggregates of Table 11.** Columns are the runs at 10^{22} training FLOPs with downstream evaluations; best value per row in bold. MMLU, Minerva MATH, and multilingual MBPP are macro averages over their subtasks (Table 14). SmolLM2 task-level scores are not archived.

Task	Natural-only $R = 0$, 21.9B	Grid $R = 0.5$, 21.9B	QA-optimal $R = 0.74$, 13.8B	Code-optimal $R = 1.0$, 9.7B
QA (BPB ↓)				
ARC-Easy	0.540	**0.490**	**0.490**	0.502
ARC-Challenge	0.736	**0.672**	0.676	0.686
HellaSwag	0.704	0.704	**0.703**	0.718
MMLU (57 subjects)	0.850	0.809	**0.801**	0.818
QA (accuracy ↑)				
ARC-Easy	0.806	**0.818**	0.809	0.798
ARC-Challenge	0.533	**0.566**	0.561	0.516
HellaSwag	**0.738**	0.725	0.733	0.702
MMLU (57 subjects)	**0.449**	0.444	0.442	0.433
Math (BPB ↓)				
GSM8K-Platinum	0.825	0.450	0.446	**0.432**
Minerva MATH (7 subjects)	1.191	0.541	0.508	**0.499**
TheoremQA	**2.794**	3.140	3.201	3.162
Code (BPB ↓)				
HumanEval (3-shot)	0.822	0.503	**0.446**	0.446
MBPP (3-shot)	1.130	0.679	0.635	**0.619**
Multilingual MBPP (17 languages)	1.175	0.521	0.480	**0.456**

N.6 Scaling Setup Ablations

N.6.1 Alternative Surface Specifications

We compare three nested surface models on the 583 runs of Appendix M. M1 shares the compute-optimal center across rewrite fractions and gives each fraction a floor shift. M2 adds an R-specific center and curvature. M3 fits an independent log-quadratic to every (C, R) cell. Table 16 reports the residual error and Bayesian information criterion (BIC) of each.

Nested F tests reject M1 against M2 ($p = 4.73 \times 10^{-12}$) and against M3 ($p = 7.79 \times 10^{-10}$). The R-dependent floor shifts nevertheless agree within ≈ 0.005 BPB across the three models, and M1 attains 99.3% of M3's R^2 with 15% of its parameters. We use M1's shared center.

N.7 Optimal Rewrite Fraction across Source Quality

The estimated floor-optimal rewrite fraction is smaller, but non-zero, on a stronger source corpus. We replace FineWeb-Edu with DCLM (Li et al., 2024) and sweep R at fixed N and D at training budgets of $C = 9 \times 10^{18}$ and 1.8×10^{19} FLOPs. These sweeps do not account for generation, so they estimate the floor $E(R)$ and its optimum R^*_∞ rather than the compute-optimal R^* of Section 2. DCLM is the stronger corpus: at every budget with final evaluations for both corpora, the natural-only DCLM run with the lowest Paloma macro BPB lies below its FineWeb-Edu counterpart (Figure 27, right), and DCLM has lower Paloma BPB in all 77 matched width–depth cells over seven budgets (Appendix N.7). On DCLM, $R = 0.1$ minimizes Paloma BPB at both budgets, and its gain over $R = 0$ is smaller than the gain of the best rewrite fraction over $R = 0$ on the FineWeb-Edu sweeps at the same budgets (Figure 27, left). The rest of this appendix fits a surface to DCLM and reports its estimated floor-optimal rewrite fraction and the matched comparison with FineWeb-Edu.

The DCLM sweep gives a fitted $\widehat{R}^*_\infty \approx 0.08$ (Figure 27). This appendix gives the natural-only ladder, the floor fit, and the matched comparison with FineWeb-Edu.

Table 14: **Minerva MATH subjects and multilingual MBPP languages** (BPB, lower is better) for the runs of Table 13; best value per row in bold.

Subtask	Natural-only $R = 0$, 21.9B	Grid $R = 0.5$, 21.9B	QA-optimal $R = 0.74$, 13.8B	Code-optimal $R = 1.0$, 9.7B
Minerva MATH				
Algebra	1.215	0.557	0.524	**0.515**
Counting and probability	1.078	0.520	0.496	**0.492**
Geometry	1.132	0.551	0.517	**0.509**
Intermediate algebra	1.324	0.589	0.544	**0.530**
Number theory	1.101	0.604	0.574	**0.566**
Prealgebra	1.039	0.494	0.472	**0.469**
Precalculus	1.446	0.473	0.431	**0.415**
Multilingual MBPP				
Bash	1.422	0.675	0.589	**0.564**
C	1.037	0.392	0.367	**0.346**
C++	1.011	0.385	0.353	**0.336**
C#	1.002	0.324	0.295	**0.282**
Go	1.366	0.495	0.435	**0.410**
Haskell	0.965	0.654	0.606	**0.586**
Java	0.985	0.283	0.262	**0.251**
JavaScript	1.170	0.469	0.440	**0.415**
MATLAB	1.067	0.545	0.507	**0.490**
PHP	1.143	0.439	0.419	**0.391**
Python	1.130	0.679	0.635	**0.619**
R	1.279	0.589	0.547	**0.542**
Ruby	1.278	0.682	0.638	**0.602**
Rust	1.458	0.650	0.580	**0.521**
Scala	1.222	0.650	0.603	**0.565**
Swift	1.256	0.479	0.446	**0.420**
TypeScript	1.180	0.470	0.430	**0.411**

Table 15: Fitted break-even allocations against SmolLM2-Instruct. Paloma crosses at 2.53×10^{22} FLOPs with $\widehat{R}^* = 0.325$, using matched within-source aggregation and context length. QA crosses at 5.32×10^{21} FLOPs with $\widehat{R}^* = 0.739$. The Math and Code crossings lie at the $R = 1$ boundary and beyond the measured ladders.

Metric	Reference	C_{total} (FLOPs)	C (FLOPs)	N^* (B)	D^* (B)	$\hat{R}^*$	Gen. share
Paloma	0.9716	2.528×10^{22}	2.505×10^{22}	38.960	107.145	0.325	0.94%
QA BPB	0.7090	5.316×10^{21}	4.946×10^{21}	11.254	73.245	0.739	6.96%
Math BPB	0.7074	5.962×10^{22}	5.769×10^{22}	34.113	281.855	1.000	3.23%
Code BPB	0.4256	4.274×10^{23}	4.189×10^{23}	55.993	1246.794	1.000	2.00%

Natural-only ladder. The DCLM ladder contains 81 completed $R = 0$ runs at eight budgets, 3×10^{18}, 9×10^{18}, 1.8×10^{19}, 3×10^{19}, 9×10^{19}, 1.8×10^{20}, 3×10^{20}, and 10^{22} FLOPs, with 5, 10, 15, 10, 14, 13, 13, and 1 runs (Figure 28). The DCLM runs follow the recipe of Appendix C with a reference point tuned on DCLM: a Gaussian-process search over the Hyperball learning rate, Adam learning rate, and ϵ at $d = 512$ and 2.5×10^9 tokens, with $\beta_1 = 0.9$ and $\beta_{2,0} = 0.9999$ fixed, gives $\eta_0 = 1.45 \times 10^{-2}$, $\eta_0^{\text{A}} = 3.28 \times 10^{-3}$, and $\epsilon_0 = 8.20 \times 10^{-12}$. An Approach-2 fit to the budget frontiers gives

$$L(N, D) = 0.67800 + \frac{46.677}{N^{0.26207}} + \frac{29.947}{D^{0.21364}}, \tag{36}$$

which implies $D^*(C) \propto C^{0.55091}$ and fits the frontier with RMSE 0.000989 BPB, against $\alpha = 0.20$ and $D^* \propto C^{0.46}$ on FineWeb-Edu. The single 10^{22} run has no within-budget width control and enters the fit as an anchor only.

Table 16: Surface-model ladder on the same 583 runs. BIC favors the shared-center model M1 although the richer models reduce residual error.

Model	Parameters	R^2	RMSE	BIC
M1: shared center	25	0.98304	0.00858	−5383
M2: center by R	45	0.98579	0.00785	−5359
M3: cell quadratics	165	0.99022	0.00651	−4813

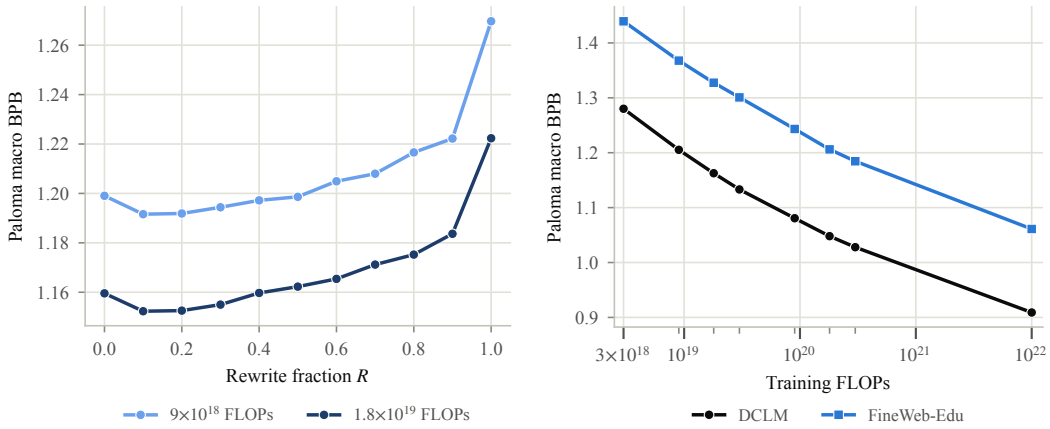

Figure 27: **Left:** Paloma macro BPB against R on DCLM at two training budgets: 550M parameters and 2.9B tokens at 9×10^{18} FLOPs, and 669M parameters and 4.7B tokens at 1.8×10^{19} FLOPs. **Right:** final Paloma macro BPB of the observed compute-optimal natural-only ($R = 0$) DCLM and FineWeb-Edu run at each training budget shown. Each point selects the lowest-Paloma run within its corpus and budget.

DCLM rewrite sweep. The sweep contains 40 runs at the two configurations of Section 3.2, one per (C, R) cell. Refitting the mixture-KL floor gives $\widehat{R}^*_\infty = 0.078$, a 0.00577-BPB gain over $R = 0$, and RMSE 0.00162 BPB.

Cross-corpus comparison. Matching DCLM runs to FineWeb-Edu $R = 0$ runs by budget, width, and depth gives 77 pairs over seven budgets. DCLM has lower Paloma BPB in all 77 pairs. The pairs are not exact: the DCLM models have 1.02–1.16× the parameters and 0.91–0.99× the tokens, and the two corpora use different tuned reference points.

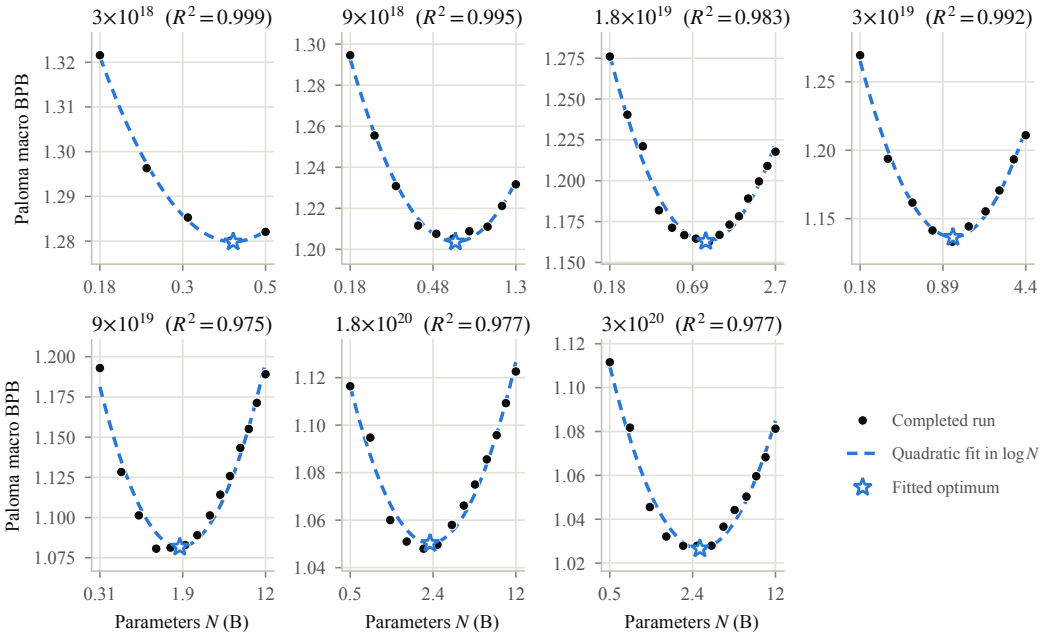

Figure 28: The seven width-controlled DCLM IsoFLOP ladders. Points are completed runs, curves are per-budget quadratics in log model size, and stars mark their minima. Each panel reports the R^2 of its quadratic. Every budget has an interior minimum over model size. The single-run 10^{22} anchor is omitted.

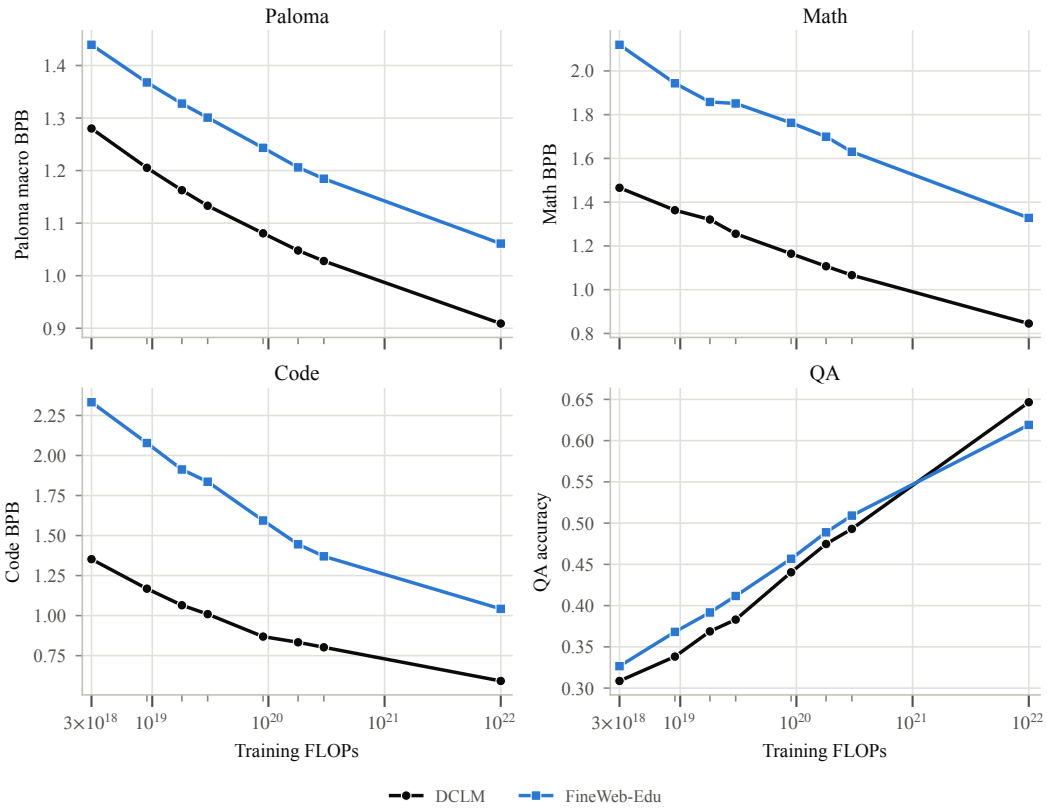

Figure 29: Final evaluations of the observed compute-optimal natural-only DCLM and FineWeb-Edu runs. At each training budget and corpus, we select the $R = 0$ run with the lowest final Paloma macro BPB and show that run's Paloma, Math, Code, and QA scores. Lower is better for the three BPB panels and higher for QA accuracy.

O QUALITATIVE REWRITE EXAMPLES

We selected the ten examples below to illustrate the four styles and the range of rewriter sizes, not to assess fidelity. Each pairs a source document with its FinePhrase rewrite and one bootstrapped-rewriter rewrite, spanning the FAQ, table, tutorial, and math styles and rewriters from 550M to 3B parameters. Several rewrites contain errors that the rewriter introduced: arithmetic mistakes in the math style, invented details in the FAQ and tutorial styles, and a contradiction of the source in one biology example. We do not measure aggregate fidelity, and the loss results in Sections 3.2 and 3.3 do not depend on it.

O.0.1 550M: FAQ

Checkpoint: `delphi-glm-550m-sft-llama3-lr1e-04-ep15`
Document: `<urn:uuid:f719a404-47c8-4aab-bc52-a5ce965f66dc>`

Original.

```
What does NEPAD stand for?
The New Partnership for Africa's Development.
What are the origins of NEPAD?
The NEPAD strategic framework document was prepared by the leaders of
    the five initiating states (Algeria, Egypt, Nigeria, Senegal, South
    Africa), in response to a mandate given to them by the Summit of the
    Organisation of African Unity (OAU). The 37th Summit of the OAU in
    July 2001 formally adopted the strategic framework document.
What is NEPAD?
NEPAD is a VISION and STRATEGIC FRAMEWORK FOR AFRICA'S RENEWAL.
NEPAD is a programme of the African Union (AU), adopted in 2001 by
    African leaders, with the primary objectives of poverty eradication,
    promotion of sustainable growth and development, and the empowerment
    of women through building genuine partnerships at country, regional
    and global levels.
NEPAD is a blueprint for Africa's development in the 21st century and
    provides unique opportunities for Africa to address the critical
    challenges facing the continent, including the attainment of the
    Millennium Development Goals (MDGs) and other continental and
    international agreed upon goals.
Why is there a need for NEPAD?
NEPAD is designed to address the current challenges facing the African
    continent. Issues such as the escalating poverty levels and
    underdevelopment of African countries and the continued
    marginalisation of Africa from the global economy needed a radical
    new intervention, spearheaded by African leaders - that would bring
    forth a new Vision for Africa's Renewal.
What is the NEPAD programme of action?
It is a detailed action plan derived from the NEPAD Strategic Framework
    document and the NEPAD Initial Action Plan, adopted by the African
    Union Summit in Durban in June 2002. The NEPAD Programme of Action
    is a holistic, comprehensive and integrated sustainable development
    initiative for the revival of Africa.
What are the NEPAD primary objectives?
What are the principles of NEPAD?
What are the elements of NEPAD's strategic focus?
In sum, NEPAD calls for adapted policy reforms and increased investments
    in the areas of agriculture and food security, science and
    technology, environment, trade and market access, governance,
    infrastructure (energy, transport and water sanitation, and
    information and communication technologies), gender and capacity
    development.
How is the NEPAD programme governed?
The work of the NEPAD Planning and Coordinating Agency - which is tasked
    with the implementation of the NEPAD Programme - is overseen by the
    NEPAD Heads of State and Government Orientation Committee (HSGOC)
    and a Steering Committee.
```

```
The Chairperson of the African Union Commission exercises supervisory
    authority over the Agency.
Basically, the main features of the AU/NEPAD governance structure
    include the African Union Assembly of Heads of State and Government
    (AU Assembly); NEPAD Heads of State and Government Orientation
    Committee (HSGOC) and a Steering Committee.
What is the NEPAD Planning and Coordinating Agency (NEPAD Agency)?
The NEPAD Agency was established by the 14th African Union Summit
    decision as the institutional vehicle for implementing the African
    Union Development agenda. Designated as the technical body of the
    African Union, the core mandate of the NEPAD Agency is to facilitate
    and coordinate the implementation of regional and continental
    priority programmes and projects and to push for partnerships,
    resource mobilisation and research and knowledge management.
```

FinePhrase.

```
NEPAD stands for New Partnership for Africa's Development, a program of
    the African Union created by the leaders of five initiating states
    (Algeria, Egypt, Nigeria, Senegal, South Africa) in 1999 and
    officially adopted by the 37th Summit of the OAU in 2001. The NEPAD
    strategic framework document addresses critical issues including
    poverty, sustainable growth, and enhancing women's empowerment
    through building genuine partnerships at all levels. It is aimed at
    achieving Millennium Development Goals and global, continental, and
    international agreed upon goals.
```

550M rewrite.

```
# Frequently Asked Questions: NEPAD, VISION, and STRATEGY for African
    Development

## Foundational Concepts

**What is NEPAD?**
The New Partnership for Africa's Development (NEPAD) is a global
    partnership designed for African development. The name stands for
    New Partnership for Africa's Development, and it defines the
    rationale for NEPAD's interventions and initiatives.

**What are the origins of NEPAD?**
The strategic framework document was prepared by five initiating states
    (Algeria, Egypt, Nigeria, Senegal, South Africa), who responded to a
    mandate given to them by the Summit of the Organisation of African
    Unity (OAU). It was signed by leaders of the five founding states
    and was officially adopted by the Summit in July 2001.
```

O.0.2 837M: TABLE

Checkpoint: `delphi-glm-837m-sft-llama3-lr2e-04-ep10`
Document: `<urn:uuid:ccdd3db1-ea0b-4b2b-885f-f0b5243ec8b3>`

Original.

```
"Predator" bacteria (green) surround "prey" bacteria (red) in this petri
    dish version of the Serengeti. Rather than eating their prey,
    however, predator cells release a chemical that activates a suicide
    gene in the prey. Prey cells also release a chemical, but one that
    promotes survival of the predators. Researchers genetically
    programmed the cells to "communicate" with each other in this way
    and function as a synthetic ecosystem. The artificial system acts as
```

```
    an experimental model and can help us understand behaviors in more
    complex, natural ecosystems. July 9, 2008
Courtesy of Hao Song, Duke University.
Full Story:
    (http://publications.nigms.nih.gov/computinglife/predator.htm)
```

FinePhrase.

```
| Cell Type | Predator | Prey |
|-----------|----------|-------|
| Green     | Red          | No            |
| 1 Death   | 1 Survival | 0             |
| 1 Death   | 0 Deactivation| 1             |
| Deactivates| Deactivates| Deactivates |

Question: What are the potential advantages and disadvantages of
    creating an engineered synthetic ecosystem?

Answer: The potential advantages include gaining insights into complex
    behaviors in ecosystems, testing hypotheses, and developing more
    efficient methods for bioremediation and biotechnology. However, the
    disadvantages may comprise risks in unintended consequences of
    introducing alien life forms, unforeseen side effects on native
    ecosystems, and challenges in long-term sustainability and control
    of the engineered ecosystem.
```

837M rewrite.

```
| Aspect | Details |
|--------|---------|
| **Topic** | Predator bacteria (green) surrounded "prey" bacteria (red)
    in a petri dish |
| **Purpose** | Study behavior of predators in artificial ecosystems |
| **Technique** | Involves releasing a chemical that activates a suicide
    gene in prey |
| **Cell Design** | Cells communicate via suicidal gene expression |
| **Experimental Setup** | Petri dish with prey and predator cells |
| **Key Finding** | Predator cells release chemical that activates
    suicide gene |
| **Research Significance** | Helps understand natural ecosystem
    behaviors |
| **Application** | Understanding natural ecosystem behavior in complex
    environments |

**Question:** How does the "Predator" bacteria's suicide gene function
    contribute to its ecosystem-based behavior?

**Answer:** The suicide gene in question functions as a chemical that
    activates a suicide gene in the prey cells, but the chemical itself
    promotes survival of the predators rather than suicide. This dual
    role allows the bacteria to both eat prey cells and release a
    chemical that promotes their survival, promoting ecosystem
    sustainability.
```

O.0.3 1B: TABLE

Checkpoint: `delphi-glm-sft-llama3-lr1e-04-ep15`
Document: `<urn:uuid:48cb395a-8ea3-4ee8-a473-4e98f269603f>`

Original.

```
Definition from Wiktionary, the free dictionary
Alternative forms
- acousticks (obsolete)
Audio (US) (file)
See -ics regarding the treatment of such nouns as singular.
- The physical quality of a space for performing music.
- (physics) The science of sounds, teaching their nature, phenomena and
    laws.
- Acoustics, then, or the science of sound, is a very considerable
    branch of physics. – Sir John Herschel.
Usage notes
- The science was previously divided by some writers into diacoustics,
    which explains the properties of sounds coming directly from (sic!
    Webster) the ear; and catacoustica, which treats of reflected sounds
    or echoes. This division is now obsolete.
Related terms
quality of a space for doing music
physics: a science of sounds
- The translations below need to be checked and inserted above into the
    appropriate translation tables, removing any numbers. Numbers do not
    necessarily match those in definitions. See instructions at Help:How
    to check translations.
```

FinePhrase.

```
**Table:**

| **Subject** | **Description**                        |
| ----------- | -------------------------------- |
| Audio         | Defined as the collection of sounds |
| that a man can hear | in the universe. In this sense    |
|  ----------- | -------------------------------- |
| Physics      | The scientific study of motion    |
|  ----------- | -------------------------------- |
| Sound        | The vibration which occurs when    |
| a sound wave happens | objects of sufficient mass    |
|  ----------- | -------------------------------- |
| Reflected    | The echo produced when a sound    |
| wave bounces back     | from a reflective surface   |
|  ----------- | -------------------------------- |
| Catacoustics | A branch of physics dealing     |
|  with reflected sounds or echoes    |

**Question and Answer Pair:**

## Question:
What branch of physics deals with the study of reflections or echoes in
    space, leading to the concept of acoustics?

## Answer:
Acoustics.
```

1B rewrite.

```
| Aspect | Details |
|:-------|:--------|
| Definition | The physical quality of a space for performing music |
| Alternative forms | Acoustics (obsolete) |
| Audio (US) | File |
| Definition | The science of sounds, teaching their nature, phenomena,
    and laws |
```

```
| Usage | They were previously divided by some writers into diacoustics;
    explains properties of sounds coming from the ear |
| Related terms | Quality of a space for doing music |
| Definition | The science of sounds, teaching their nature, phenomena,
    and laws |
| Usage | Were previously divided by some writers into diacoustics;
    explains division now obsolete |
| Related terms | Quality of a space for doing music |
| Definition | The science of sounds, teaching their nature, phenomena,
    and laws |
| Usage | Were formerly divided by some writers into diacoustics;
    explains division now obsolete |
| Related terms | Quality of a space for doing music |

**Question:** How does the usage note for "sound" differ from how it was
    historically defined?

**Answer:** The usage note states that "sound" was previously divided
    into diacoustics, explains that this division is now obsolete, and
    references "quality of a space for doing music" as a related term.
```

O.0.4 1B: FAQ

Checkpoint: `delphi-glm-sft-llama3-lr1e-04-ep15`
Document: `<urn:uuid:4fd3037a-b829-4827-8fa6-c3aeb3e57890>`

Original.

```
Pain Management and Bioelectric Therapy
- Introduction to bioelectric therapy
- What conditions are treated with bioelectric therapy?
- How effective is bioelectric therapy for pain relief?
- What happens during bioelectric therapy?
- What are the side effects of bioelectric therapy?
- How often should people get bioelectric therapy?
- How do people prepare for bioelectric therapy?
Introduction to Bioelectric Therapy
Bioelectric therapy is a safe, drug-free treatment option for people in
    pain. It is used to treat some chronic pain and acute pain
    conditions. It relieves pain by blocking pain messages to the brain.
    When you are injured, pain receptors send a message to the central
    nervous system (the brain and spinal cord). The message is
    registered as pain by certain cells in the body. Using bioelectric
    currents, bioelectric therapy relieves pain by interrupting pain
    signals before they reach the brain. Bioelectric therapy also
    prompts the body to produce endorphins which help to relieve pain.
What Conditions Are Treated With Bioelectric Therapy?
Bioelectric therapy can be used to treat chronic and acute pain
    conditions including:
- Complex regional pain syndrome, also known as reflex sympathetic
    dystrophy or RSD.
- Back pain.
- Muscle pain.
- Headaches and migraines.
- Disorders of blood flow in the upper and lower limbs.
- Temporomandibular joint (TMJ) syndrome (which affects the jaw).
- Disorders of the nervous system, such as diabetic neuropathy.
- Pain and ulcers of the skin resulting from poor circulation or
    scleroderma (a chronic condition that can cause thickening or
    hardening of the skin).
Bioelectric therapy isn't right for everyone. It is not recommended for
    people who:
- Have a pacemaker.
```

```
- Are pregnant.
- Have thrombosis (blood clots in the arms or legs).
- Have a bacterial infection.
Chronic Pain/Back Pain
Find tips and advances in treatment.
```

FinePhrase.

```
Bioelectric therapy is a non-invasive, drug-free treatment that can help
    manage chronic pain and back pain, reduce inflammation, and speed up
    the healing process.
```

1B rewrite.

```
# Frequently Asked Questions: Pain Management and Bioelectric Therapy

## Understanding Bioelectric Therapy

**What is bioelectric therapy?**
Bioelectric therapy is a safe, drug-free treatment option for people in
    pain. It is used to treat some chronic pain and acute pain
    conditions. It relieves pain by blocking pain messages to the brain.

**What conditions does bioelectric therapy treat?**
Bioelectric therapy treats some chronic pain conditions and acute pain
    conditions. It relieves pain by blocking pain messages to the brain.

**How effective is bioelectric therapy for pain relief?**
Bioelectric therapy is highly effective for pain relief. It works by
    interrupting pain signals before they reach the brain, rather than
    stimulating the brain to produce morphine or endorphins themselves.

**What happens during bioelectric therapy?**
During bioelectric therapy, electrodes are placed on the skin to deliver
    electrical current. This current then passes through the body,
    interrupting pain signals that would otherwise reach the brain. The
    affected area is then exposed to the therapeutic bioelectric
    currents.

## Side Effects and Precautions

**What are the side effects of bioelectric therapy?**
While most bioelectric therapies are safe and effective, some side
    effects can occur. These can include skin reactions, infections, and
    potential injury to the brain or nervous system.

**How often should people get bioelectric therapy?**
People who receive bioelectric therapy should take it as directed to
    ensure proper pain relief.

## Treatment Duration

**When should people get bioelectric therapy?**
People who receive bioelectric therapy should take it as directed
    because it interrupts pain signals before they reach the brain. The
    treatment also prompts the body to produce endorphins, which help to
    relieve pain.

## Special Considerations

**Are bioelectric therapy and pain management exactly the same thing?**
No. While they share many treatments, bioelectric therapy and pain
    management are distinct. Bioelectric therapy uses electrical
```

```
    currents to block pain messages, while pain management uses
    medication to manage the pain.

**Who should not take bioelectric therapy?**
People who have a pacemaker should not take bioelectric therapy. People
    who are pregnant should also avoid bioelectric therapy. These
    peoples' bodies may not react to the electrical current as it passes
    through them.

## Additional Resources

**Where can I find more information about bioelectric therapy?**
Additional information can be found in the resources for pain management
    and bioelectric therapy. Topics include history, types of pain
    conditions treated, effective treatment methods, side effects, and
    safety concerns.
```

O.0.5 3B: TABLE

Checkpoint: `delphi-glm-3b-sft-llama3-lr5e-05-ep15`
Document: `<urn:uuid:3e09eef3-39c3-41eb-a404-bb93536daa7f>`

Original.

```
When Your Child Has Pharyngitis or Tonsillitis
|Pain or redness can occur in these areas.|
What Causes Pharyngitis or Tonsillitis?
Most commonly, pharyngitis and tonsillitis are caused by a viral or
    bacterial infection.
What Are the Symptoms of Pharyngitis or Tonsillitis?
The main symptom of both conditions is a sore throat. Your child may
    also have a fever, redness or swelling of the throat, and trouble
    swallowing.
How Is Pharyngitis or Tonsillitis Diagnosed?
The healthcare provider will examine your child's throat. The healthcare
    provider might swab (wipe) your child's throat. This swab will be
    tested for the bacteria that causes an infection called strep
    throat. If needed, a blood test can be done to check for a viral
    infection, such as mononucleosis.
How Is Pharyngitis or Tonsillitis Treated?
If your child's sore throat is caused by a bacterial infection, the
    healthcare provider may prescribe antibiotics. Otherwise, you can
    treat your child's sore throat at home. To do this:
Give your child acetaminophen or ibuprofen to ease the pain. Don't give
    your child aspirin to relieve a fever. Using aspirin to treat a
    fever in children could cause a serious condition called Reye's
    syndrome.
Give your child cool liquids to drink.
Have your child gargle with warm saltwater if it helps relieve pain. An
    over-the-counter throat numbing spray may also help.
What Are the Long-Term Concerns?
If your child has frequent sore throats, take him or her to see a
    healthcare provider. Removing the tonsils may help relieve your
    child's recurring problems.
Call your child's healthcare provider right away if your child has any
    of the following:
```

FinePhrase.

```
|Heading | Description |
|---------|-------------|
|When Your Child Has Pharyngitis or Tonsillitis|
|What to Know About Pharyngitis or Tonsillitis|
|Causes of Pharyngitis or Tonsillitis|
|Symptoms of Pharyngitis or Tonsillitis|
|Diagnosis Process for Pharyngitis or Tonsillitis|
|How to Treat Pharyngitis or Tonsillitis|
|Long-Term Concerns|

What Are the Long-Term Concerns for Pharyngitis or Tonsillitis?
```

3B rewrite.

```
| Category | Details |
|----------|---------|
| **Condition** | Pharyngitis/tonsillitis |
| **Cause** | Viral or bacterial infection |
| **Symptoms** | Sore throat, fever, redness/swelling, difficulty
    swallowing |
| **Key Symptom** | Sore throat |
| **Alternative Symptoms** | Fever, redness/swelling, swallowing
    problems |
| **Diagnosis** | Medical provider examines throat for strep throat;
    blood test for viral infection (mononucleosis); or tonsil removal if
    needed |
| **Treatment** | Antibiotics for bacterial infections; acetaminophen or
    ibuprofen for pain; cool liquids for sore throat; gargling with
    saltwater for swallowing |
| **Self-Care Steps** | Remove tonsils if sore throat is caused by
    bacteria; avoid aspirin for fever; use throat numbing spray if
    needed; avoid throat drying |
| **Long-Term Concerns** | Frequent sore throats; removal of tonsils for
    recurring problems |

**Question:** Why is removing the tonsils recommended for children with
    recurring pharyngitis or tonsillitis?

**Answer:** Removing the tonsils helps relieve children's recurring
    problems because the tonsils serve as a reservoir for bacteria, and
    removing them creates a clear path for healing.
```

O.0.6 3B: TUTORIAL

Checkpoint: `delphi-glm-3b-sft-llama3-lr5e-05-ep15`
Document: `<urn:uuid:66166475-112f-43f3-a5c1-215ae0e2a518>`

Original.

```
Now more than five decades old, Herbert Hoover's 1959 statement to the
    Board of Trustees of Stanford University on the purpose and scope of
    the Hoover Institution continues to guide and define its mission in
    the twenty-first century:
"This Institution supports the Constitution of the United States, its
    Bill of Rights and its method of representative government. Both our
    social and economic systems are based on private enterprise from
    which springs initiative and ingenuity.... Ours is a system where
    the Federal Government should undertake no governmental, social or
    economic action, except where local government, or the people,
    cannot undertake it for themselves.... The overall mission of this
    Institution is, from its records, to recall the voice of experience
```

```
    against the making of war, and by the study of these records and
    their publication, to recall man's endeavors to make and preserve
    peace, and to sustain for America the safeguards of the American way
    of life. This Institution is not, and must not be, a mere library.
    But with these purposes as its goal, the Institution itself must
    constantly and dynamically point the road to peace, to personal
    freedom, and to the safeguards of the American system."
The principles of individual, economic, and political freedom; private
    enterprise; and representative government were fundamental to the
    vision of the Institution's founder. By collecting knowledge,
    generating ideas, and disseminating both, the Institution seeks to
    secure and safeguard peace, improve the human condition, and limit
    government intrusion into the lives of individuals.
```

FinePhrase.

```
I. Introduction
- A brief overview of the importance of the Hoover Institution
- The mission and purpose of the institution
II. Numbered Steps to Understand the Institution's Mission
III. Understanding Individual Freedom
   - What it means to be a responsible citizen
IV. The Role of Government in Society
   - Providing public goods
   - Encouraging innovation
V. Education and Its Importance
   - The value of knowledge in personal and societal development
VI. A Life of Liberty
   - Fostering creativity
   - Promoting self-reliance
VII. Conclusion
- Recap of the institution's principles
- Importance of individual freedom, economic freedom, political freedom,
    and preserving personal freedom
VIII. Further Resources
- The official website of the Hoover Institution
- Other sources of information on the institution
```

3B rewrite.

```
# Understanding Herbert Hoover's 1959 Statement on the Purpose and Scope
    of the Hoover Institution: A Tutorial

## Background and Context

- **Date:** December 5, 1959
- **Historical context:** Established just over five decades earlier
- **Key phrase:** "...support the Constitution of the United States, its
    Bill of Rights, and method of representative government"
- **Historical context:** These principles guided the Institute's
    mission through the twentieth century.

## Step 1: Review the Institution's Founding Principles

The Hoover Institution operates under the following principles:

1. **Individual freedom:** Ensures individual rights and freedoms are
    protected
2. **Economic freedom:** Supports private enterprise through free
    inquiry and knowledge generation
3. **Political freedom:** Supports representative government through the
    ballot box and public debate
```

```
4. **Respect for law:** Guarantees legal authority and institutional
    integrity
5. **Sustaining American Way of Life:** Emphasizes local
    self-determination and sustainable practices

## Step 2: Understand the Institution's Mission

The mission of the Hoover Institution is:

- **Recollect the voice of experience against the making of war**
- **Recollect man's efforts to make and preserve peace**
- **Sustain for America the safeguards of the American way of life**

## Step 3: Recognize the Institute's Goal

The mission statement sets the foundation for action:

- Preventing war
- Improving human conditions
- Limiting government intrusion into individual lives

## Step 4: Understand the Role of the Board

The Board of Trustees continues to define the Institution's purpose:

- **Focus:** Today, through leadership and expertise
- **Mission:** Keep the Institution's mission clear through consistent
    consideration and action

## Step 5: Access the Institute's Records

To support the mission, the Board of Trustees maintains:

- **Scope:** The records of the Hoover Institution
- **Scope:** They collect knowledge, generate ideas, and disseminate
    both for educational and scholarly purposes
- **Purpose:** Secure and safeguard peace, improve the human condition,
    and limit government intrusion into individual lives

## Step 6: Apply These Principles in Daily Life

- **Individual freedom:** Protect the rights and freedoms you enjoy
- **Economic freedom:** Support private enterprise through free
    enterprise
- **Political freedom:** Support representative government through the
    ballot box and public debate
- **Respect for law:** Ensure legal authority and institutional integrity
- **Sustaining the American way of life:** Emphasize local
    self-determination and sustainable practices

## Additional Resources

- Source: Herbert Hoover's 1959 Statement on the Purpose and Scope of
    the Hoover Institution: A Tutorial
- Source: Washington, D.C. (December 5, 1959) - Herbert Hoover's 1959
    Annual Report, Stanford University
- Source: http://www.hoover.org/archive/statement/ (accessed December 5,
    2019)
```

O.0.7 1B: MATH

Checkpoint: `delphi-glm-sft-llama3-lr1e-04-ep15`
Document: `<urn:uuid:63c9a260-1d93-4db5-bbb6-9126d821ed64>`

Original.

```
ZomeTools - Zome Systems
The ZomeTools - Zome System
, is a fantastic Geometry Education system. Made of precision molded ABS
    plastic, the ZomeTools Geometry kits are versatile manipulative
    tools for problem solving and hands-on exploration of math, geometry
    and science concepts for first graders through grad students. Color
    and shape coded parts integrate the concepts of shape, number and
    vector into one coherent geometry system. The result is the best
    educational geometry construction system on the market. Zome System
    / ZomeTools components are designed to incorporate the Fibonacci
    sequence and the Golden Proportion - the very same proportions that
    govern the growth of cells, crystals and other natural forms.
connector node provides 62 buildable directions in space. This allows
    thousands of geometrical Zome structures
that can't be built in any other system.
With the help of Zometools you can continue to explore: Why so many
    everyday items like balls, pizzas, and can lids are round. The hows
    and whys of the development of our number system by the ancient
    Greeks. Why geometric shapes show up so often in art, science and
    board games. What properties attributed to the spiral make it the
    most widespread shape in nature, from embryos and hair curls to
    hurricanes and galaxies. How the human body shares the design of a
    corn plant and the solar system. How a snowflake is like Stonehenge,
    and a beehive like a calendar. How our ten fingers hold the secrets
    of both a lobster and a cathedral. Zometools provide an introduction
    to the geometric code of nature and the universe.
```

FinePhrase.

```
Problem:

Jenna found 2 ZomeTools sets from ZomeTools at the store. She bought the
    total amount of sets so she bought 10 sets. Then, she traded 5 sets
    with Daniel for StarLego. After that, she spent $20 to buy more
    ZomeTools sets. If Jenna has 2 ZomeTools sets left, how much did
    each ZomeTools set cost her after purchasing from ZomeTools and
    trading?
```

1B rewrite.

```
**Problem:**

A teacher is building a geometric ZomeTools kit for her first-grade
    class. The teacher has 62 buildable directions in space, and each
    direction contains 373 built-in directions. She wants to provide her
    students with access to as many buildable directions as possible,
    noting that 1 direction contains 5 built-in directions. If she
    allocates 4 sections of space per direction, how many total built-in
    directions will the class have access to?

**Solution:**

Step 1: Identify the total number of built-in directions.
Total directions = 62

Step 2: Calculate the number of built-in directions available.
```

```
62 x 5 = 252 built-in directions

Step 3: Determine how many of these directions can be used.
252 x 5 = 258 directions

Step 4: Calculate the remaining number of directions to be used.
258 - 258 = 48 directions

Step 5: Determine how many students can access as many built-in
    directions as possible.
48 x 5 = 144 students

Answer: The class will have 258 built-in directions available, and 144
    students will be able to access as many of these directions as
    possible.
```

O.0.8 3B: MATH

Checkpoint: `delphi-glm-3b-sft-llama3-lr5e-05-ep15`
Document: `<urn:uuid:d29bb74f-45d0-452c-b599-d774fedf25bd>`

Original.

```
Colored map compiled by merging geologic and topographic maps of the
    United States through computer processing and enhancement with links
    to panoramic movie using QuickTime, descriptions of features, and a
    game entitled Puzzle of Regions.
The USGS provides map-on-demand (MOD) printing for selected infrequently
    requested geologic and other theme map products providing
    high-quality paper-copy maps as needed. Site links to available map
    products with thumbnail views and how to order.
This publication (Open-File Report 95-479) is a teacher's guide to two
    U.S. Geological Survey maps that show gravity and magnetic data in
    contour form superimposed on a LANDSAT satellite image of the San
    Francisco Bay area.
Techniques of computer-based representations of areal geology to
    construct three-dimensional geologic maps that retain all the
    information of traditional geologic maps while providing a more
    detailed picture of the subsurface.
Main page for accessing links for information and data on the San
    Francisco Bay estuary and its watershed with links to highlights,
    water, biology, wetlands, hazards, digital maps, geologic mapping,
    winds, bathymetry and overview of the Bay.
```

FinePhrase.

```
What is the value of dividing 50 by 2 x 3 and then multiplying this
    result by 15?
```

3B rewrite.

```
**Problem:**

A geological map shows geologic formations and topographic features of
    the United States. The map shows 24 different types of geologic
    formations and 5 different topographic zones. Additionally, the map
    names 3 major mountain systems and 4 major drainage systems. If the
    map's publisher requires a teacher's guide to 2 U.S. Geological
    Survey maps that show gravity and magnetic data in contour form
    superimposed on a LANDSAT satellite image of the San Francisco Bay
    area, and the 2nd map costs $24 per copy, how many total copies of
```

```
    the teacher's guide must the publisher order to sell enough copies
    to cover the total cost of the 2 maps?

**Solution:**

Step 1: Identify the number of different geologic formations on the 24
    types of formations map.
Total formations = 24

Step 2: Identify the number of different topographic zones on the 5
    different topographic zones map.
Topographic zones = 5

Step 3: Calculate the total number of formations and topographic zones
    on the 24 types of formations map.
24 types x 3 formations per type = 72 formations and topographic zones

Step 4: Calculate the total number of copies of the 24 types of
    formations map needed.
72 formations x 2 copies per formation = 240 copies

Step 5: Calculate the total cost of the 2 maps.
240 copies x $24 per copy = $720

The publisher must order 720 copies of the teacher's guide to sell
    enough copies to cover the total cost.
```

O.0.9 1B: TUTORIAL

Checkpoint: `delphi-glm-sft-llama3-lr1e-04-ep15`
Document: `<urn:uuid:66166475-112f-43f3-a5c1-215ae0e2a518>`

Original.

```
Now more than five decades old, Herbert Hoover's 1959 statement to the
    Board of Trustees of Stanford University on the purpose and scope of
    the Hoover Institution continues to guide and define its mission in
    the twenty-first century:
"This Institution supports the Constitution of the United States, its
    Bill of Rights and its method of representative government. Both our
    social and economic systems are based on private enterprise from
    which springs initiative and ingenuity.... Ours is a system where
    the Federal Government should undertake no governmental, social or
    economic action, except where local government, or the people,
    cannot undertake it for themselves.... The overall mission of this
    Institution is, from its records, to recall the voice of experience
    against the making of war, and by the study of these records and
    their publication, to recall man's endeavors to make and preserve
    peace, and to sustain for America the safeguards of the American way
    of life. This Institution is not, and must not be, a mere library.
    But with these purposes as its goal, the Institution itself must
    constantly and dynamically point the road to peace, to personal
    freedom, and to the safeguards of the American system."
The principles of individual, economic, and political freedom; private
    enterprise; and representative government were fundamental to the
    vision of the Institution's founder. By collecting knowledge,
    generating ideas, and disseminating both, the Institution seeks to
    secure and safeguard peace, improve the human condition, and limit
    government intrusion into the lives of individuals.
```

FinePhrase.

```
I. Introduction
- A brief overview of the importance of the Hoover Institution
- The mission and purpose of the institution
II. Numbered Steps to Understand the Institution's Mission
III. Understanding Individual Freedom
   - What it means to be a responsible citizen
IV. The Role of Government in Society
   - Providing public goods
   - Encouraging innovation
V. Education and Its Importance
   - The value of knowledge in personal and societal development
VI. A Life of Liberty
   - Fostering creativity
   - Promoting self-reliance
VII. Conclusion
- Recap of the institution's principles
- Importance of individual freedom, economic freedom, political freedom,
    and preserving personal freedom
VIII. Further Resources
- The official website of the Hoover Institution
- Other sources of information on the institution
```

1B rewrite.

```
# Understanding the Herbert Hoover Institution: A Guide to Its Legacy
    and Core Principles

## Background and Context

**Step 1: Identify the Institution's Historical Significance**
- **Established:** 1959
- **Original Purpose:** Purposeed for 5 decades to preserve the
    Institution's role in American democracy
- **Core Purpose:** Support for its Constitution, Bill of Rights, and
    method of representative government
- **Historical Context:** Founded in 1959 by President Herbert Hoover,
    this institution continues to serve America today

## Step 2: Understand the Mission Statement**
- **Core Purpose:** Recall the voice of experience against the making of
    war
- **Mission Statement:** Avoid creating unnecessary governmental,
    social, or economic actions
- **Approach:** Study the records to understand efforts to make and
    preserve peace, and by study, to illuminate the bounds of American
    way of life

## Step 3: Recognize the Historical Significance of Individual Liberty**
- **Core Principle:** The core mission of this Institution is to recall
    the voice of experience against the making of war
- **Approach:** Avoid creating unnecessary governmental, social, or
    economic actions
- **Historical Context:** Enable voluntary, collaborative engagement
    with peace
- **Emergency Response:** Acknowledge that the Institution cannot
    operate effectively without proper community support

## Step 4: Understand the Role of the Individual in American Society**
- **Core Principle:** The individual must play a meaningful role in
    shaping his or her own society
- **Approach:** Foster personal freedom and the safeguards of the
    American system as primary goals
```

```
- **Historical Context:** Remember that government intrusion into
    personal decisions is not acceptable

## Step 5: Apply Core Principles to Contemporary Challenges**
- **Purpose:** Restore the Institute's role in American democracy by:
  - Recreating the voices of experience against the making of war
  - Creating peace through research, education, and public engagement
  - Limiting government intrusion into individual lives
  - Strengthening the American way of life through individual freedom
    and limited government

## Key Takeaway
The Herbert Hoover Institution's legacy goes beyond preserving
    literature. By creating a space for collective knowledge, learning,
    and collaborative engagement, it seeks to secure and safeguard
    peace, improve the human condition, and limit government intrusion
    into individual lives.
```

O.0.10 3B: FAQ

Checkpoint: `delphi-glm-3b-sft-llama3-lr5e-05-ep15`
Document: `<urn:uuid:4fd3037a-b829-4827-8fa6-c3aeb3e57890>`

Original.

```
Pain Management and Bioelectric Therapy
- Introduction to bioelectric therapy
- What conditions are treated with bioelectric therapy?
- How effective is bioelectric therapy for pain relief?
- What happens during bioelectric therapy?
- What are the side effects of bioelectric therapy?
- How often should people get bioelectric therapy?
- How do people prepare for bioelectric therapy?
Introduction to Bioelectric Therapy
Bioelectric therapy is a safe, drug-free treatment option for people in
    pain. It is used to treat some chronic pain and acute pain
    conditions. It relieves pain by blocking pain messages to the brain.
    When you are injured, pain receptors send a message to the central
    nervous system (the brain and spinal cord). The message is
    registered as pain by certain cells in the body. Using bioelectric
    currents, bioelectric therapy relieves pain by interrupting pain
    signals before they reach the brain. Bioelectric therapy also
    prompts the body to produce endorphins which help to relieve pain.
What Conditions Are Treated With Bioelectric Therapy?
Bioelectric therapy can be used to treat chronic and acute pain
    conditions including:
- Complex regional pain syndrome, also known as reflex sympathetic
    dystrophy or RSD.
- Back pain.
- Muscle pain.
- Headaches and migraines.
- Disorders of blood flow in the upper and lower limbs.
- Temporomandibular joint (TMJ) syndrome (which affects the jaw).
- Disorders of the nervous system, such as diabetic neuropathy.
- Pain and ulcers of the skin resulting from poor circulation or
    scleroderma (a chronic condition that can cause thickening or
    hardening of the skin).
Bioelectric therapy isn't right for everyone. It is not recommended for
    people who:
- Have a pacemaker.
- Are pregnant.
- Have thrombosis (blood clots in the arms or legs).
- Have a bacterial infection.
```

```
Chronic Pain/Back Pain
Find tips and advances in treatment.
```

FinePhrase.

```
Bioelectric therapy is a non-invasive, drug-free treatment that can help
    manage chronic pain and back pain, reduce inflammation, and speed up
    the healing process.
```

3B rewrite.

```
# Frequently Asked Questions: Pain Management and Bioelectric Therapy

## Foundations

**What is bioelectric therapy?**
Bioelectric therapy is a drug-free treatment option for people in pain.
    It uses electrical currents to interrupt pain messages before they
    reach the brain, which helps relieve pain.

**What conditions are treated with bioelectric therapy?**
Bioelectric therapy is used to treat certain chronic pain and acute pain
    conditions, including:
- Complex regional pain syndrome
- Back pain
- Muscle pain
- Headaches and migraines
- Disorders of blood flow in the upper and lower limbs
- Temporomandibular joint (TMJ) syndrome
- Disorders of the nervous system, such as diabetic neuropathy

**How does bioelectric therapy work?**
Bioelectric therapy relieves pain by blocking pain messages to the
    brain. When you are injured, pain receptors send pain messages to
    the central nervous system (the brain and spinal cord). The message
    is registered as pain by certain cells in the body. The cells
    interpret this pain as threatening and trigger the protective
    response of pain relief mechanisms.

## How Bioelectric Therapy Works

**How does bioelectric therapy relieve pain?**
Bioelectric therapy prompts the body to produce endorphins. These are
    compounds that help relieve pain. Additionally, the therapy prompts
    the brain to produce more of its own endorphins, which helps to
    further relieve the pain.

**What happens during bioelectric therapy?**
Bioelectric therapy is a non-invasive procedure that uses electrical
    currents to block pain messages before they reach the brain. The
    repeated stimulation of specific pain-relieving nerves causes the
    body produce its own endorphins, which reduce pain.

**How often should people get bioelectric therapy?**
People should receive bioelectric therapy regularly - either as an
    outpatient treatment or in certain cases, as part of a
    rehabilitation program.

## Preparation and Limitations

**Is bioelectric therapy right for everyone?**
```

```
No. Bioelectric therapy is not recommended for people who have a
    pacemaker, are pregnant, have thrombosis (blood clots in the legs or
    arms), or have a bacterial infection.

**What conditions don't respond well to bioelectric therapy?**
Bioelectric therapy is not recommended for conditions like chronic pain,
    back pain, muscle pain, or headaches and migraines that don't
    respond to other treatments.

## Additional Information

**Where can I find more information about bioelectric therapy?**
Additional information is available through the company that supplies
    the treatment equipment, or through medical websites that cover pain
    management and bioelectric therapy.
```