GRIP: GEOMETRIC REFINEMENT AND ADAPTIVE INFORMATION POTENTIAL FOR DATA EFFICIENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient language-model pre-training requires deciding how to divide a fixed token budget across semantic regions and which examples to retain within each region. We introduce **GRIP** (Geometric Refinement and Adaptive Information Potential), which combines model-conditioned allocation with length-aware geometric selection. At the cluster level, static quality and normalized held-out loss reduction determine token budgets. Within clusters, a length-rectified inverse-density sampler increases geometric coverage while retaining long documents. We evaluate 8B and 16B Mixture-of-Experts models on eight code benchmarks with budgets up to 300B tokens. Across five paired seeds at 300B tokens, GRIP exceeds UniGeM by **0.9** and **1.3 average-score points** at 8B and 16B, respectively. Across three paired seeds at 100B tokens, GRIP reaches 35.5 ± 0.3, **1.2 points** above UniGeM. The ablation separates the gains from cluster allocation and document selection.

1 INTRODUCTION

Scaling language models makes corpus composition increasingly important: noisy or redundant data consumes substantial compute without comparable gains (Kaplan et al., 2020; Hoffmann et al., 2022; Gadre et al., 2024; Goyal et al., 2024). Data curation targets useful and representative supervision under a fixed token budget (Sorscher et al., 2022; Albalak et al., 2024).

Existing methods usually optimize either cluster-level allocation (Xie et al., 2023a; Liu et al., 2025; Diao et al., 2025) or instance-level selection (Zhuang et al., 2025; Yu et al., 2024; Zhang et al., 2025b). Treating them separately misaligns the two decisions: informative clusters receive insufficient capacity, and cluster budgets are filled with redundant examples. Code corpora amplify this mismatch through diverse domains, strict syntax, and long-range dependencies (Li et al., 2023; Guo et al., 2020). UniGeM jointly addresses mixing and selection through static corpus geometry (Wang et al., 2026); GRIP adds model-conditioned difficulty to the allocation signal (Wang et al., 2025). Within clusters, anisotropy and length-induced embedding collapse make long documents appear artificially redundant (Ethayarajh, 2019; Zhou et al., 2025).

GRIP combines static cluster quality with proxy held-out loss reduction for model-conditioned allocation, then applies length-aware geometric selection within each cluster. Its Rapid Adaptation Probe measures held-out loss reduction from a controlled initialization and converts information potential into a replay multiplier. A length-rectified inverse-density rule promotes geometric coverage while preserving long documents. Allocation is computed offline before target-model training.

We train 8B and 16B MoE models from scratch. Across five paired seeds at 300B tokens, GRIP exceeds UniGeM by **0.9** and **1.3 average-score points**, respectively, and exceeds random sampling by **9.0** and **8.7 points**. At 100B tokens, three paired seeds yield 35.5 ± 0.3, **1.2 points** above UniGeM. GRIP contributes:

- **Model-conditioned allocation:** The **Rapid Adaptation Probe** combines static quality with *held-out cluster-level loss reduction*. The resulting information potential sets the replay budget for each cluster.
- **Length-aware sampling:** A **length-rectified inverse-density sampler** offsets the measured association between document length and embedding density while retaining within-cluster diversity.

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.

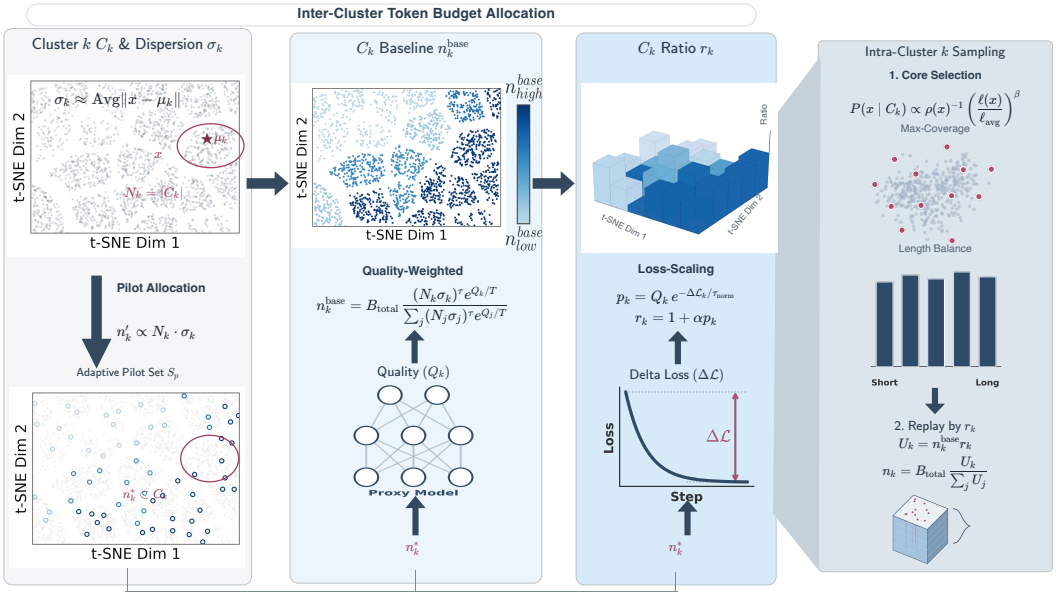


Figure 1: **Overview of GRIP.** A dispersion-aware probe estimates cluster quality and adaptation behavior. Cluster mass, dispersion, and quality define the baseline budget; information potential p_k determines the replay multiplier r_k. Global normalization produces cluster quotas, which are filled by length-aware inverse-density sampling.

- **Multi-seed scaling results:** Experiments train 8B and 16B MoE models with budgets up to 300B tokens, using five paired seeds for the main comparison and three for the ablation.

2 Method

GRIP selects data at two levels, as shown in Figure 1. **Inter-Cluster Budgeting** assigns a token budget to each semantic cluster using corpus size, geometric dispersion, static quality, and proxy-model adaptation. **Intra-Cluster Selection** then fills each budget with diverse examples while correcting for length-related density bias.

2.1 Problem Formulation

Let $\mathcal{D} = \{x_i\}_{i=1}^{N_D}$ be a corpus, and let $\ell(x)$ denote the number of tokens in sample x. GRIP constructs a subset $S \subset \mathcal{D}$ under the constraint $\sum_{x \in S} \ell(x) \leq B_{total}$. An encoder f_θ maps each sample to a normalized embedding $\mathbf{e}(x) = f_\theta(x) \in \mathbb{S}^{d-1}$. Rao's quadratic entropy summarizes the geometric diversity of a sampling distribution π (Botta-Dukát, 2005):

$$\mathcal{I}(\pi) \triangleq \sum_{x,y \in \mathrm{supp}(\pi)} \pi(x)\,\pi(y)\,d\big(\mathbf{e}(x), \mathbf{e}(y)\big), \tag{1}$$

where $d(\cdot,\cdot)$ is cosine distance. GRIP decomposes this corpus-level coverage statistic by semantic cluster: inter-cluster budgeting preserves mixture structure, and intra-cluster sampling increases local geometric coverage (Xie et al., 2023b).

2.2 Corpus Representation and Probe Construction

Representation Space and Geometric Metrics. The `Qwen/Qwen3-Embedding-8B` checkpoint supports 32K-token inputs and uses the final EOS-token state for pooling (Zhang et al., 2025c). We split document x into contiguous chunks $x_1, \ldots, x_m$ with maximum length $W = 32{,}768$. Each chunk is embedded with the official EOS pooling rule; document embeddings are the normalized,

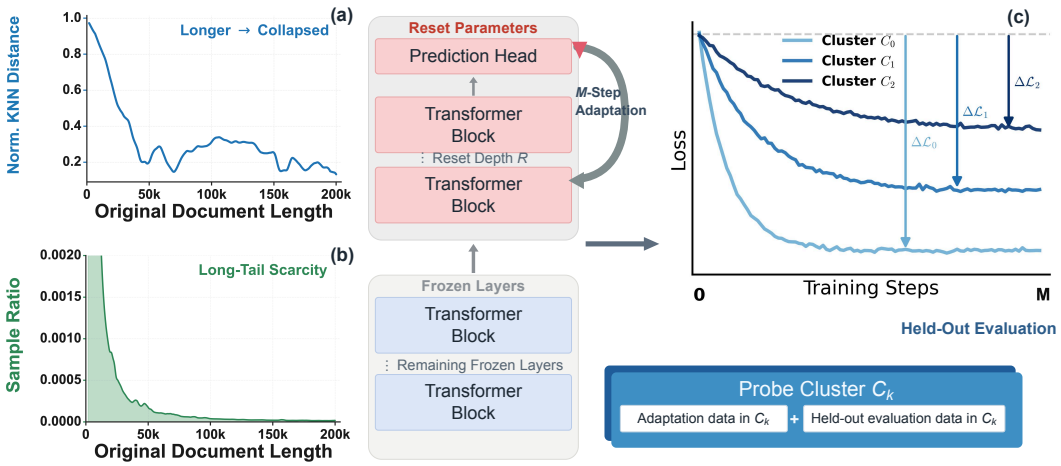


Figure 2: **Document-length bias and adaptation probing.** (a) Normalized 10-NN distance as a function of original document length after 32K chunking, EOS pooling, and document-level aggregation. (b) Distribution of original document lengths. (c) RAP resets the prediction head and optionally the upper R blocks, performs M adaptation steps, and measures $\Delta\mathcal{L}_k$ on held-out probe data; the curves plot the measured losses.

token-weighted mean of the chunk embeddings:

$$\mathbf{e}(x) = \mathrm{norm}\left(\frac{\sum_{i=1}^{m} \ell(x_i)\mathbf{e}(x_i)}{\sum_{i=1}^{m} \ell(x_i)}\right). \tag{2}$$

Lengths above 32K in our analysis refer to original document length; each encoder input remains within the 32K window. Spherical k-means partitions the document embeddings into K disjoint clusters $\mathcal{C} = \{C_1, \ldots, C_K\}$ (Johnson et al., 2021; Bachem et al., 2018). We characterize cluster C_k by its **geometric dispersion** σ_k, the root mean square distance from centroid μ_k:

$$\sigma_k = \sqrt{|C_k|^{-1} \sum_{x \in C_k} \|\mathbf{e}(x) - \mu_k\|^2}. \tag{3}$$

σ_k is small for compact clusters and large for heterogeneous clusters. It quantifies the within-cluster variation that the probe and selected subset must cover (Gao et al., 2021).

Dispersion-Aware Probe Construction. We allocate the probe set $\mathcal{P}$ in proportion to cluster mass and geometric dispersion, following Neyman allocation. Writing $N_k = |C_k|$,

$$n_k^{probe} \propto N_k \cdot \sigma_k, \quad \text{s.t.} \sum n_k^{probe} = B_{probe}. \tag{4}$$

Compact clusters receive fewer probe samples, while heterogeneous clusters receive more.

Length-Induced Density Compression. Figure 2 relates original document length to neighbor distance after the chunk-and-aggregate procedure in Eq. 2. Long documents are rare and concentrated at smaller normalized neighbor distances, causing inverse-density sampling to under-select them. Length correction counteracts this density bias.

2.3 Inter-Cluster Budgeting

The cluster budget has two terms. The *baseline budget* combines cluster size, dispersion, and static quality; the *replay multiplier* adds proxy-conditioned weighting. Both are computed before target-model pre-training.

Static Baseline Budget. The baseline allocation applies a sub-linear function to cluster mass and dispersion, then weights it by the probe-estimated quality score Q_k:

$$n_k^{base} = B_{total} \cdot \frac{(N_k \cdot \sigma_k)^\tau \cdot \exp(Q_k/T)}{\sum_{j=1}^{K} (N_j \cdot \sigma_j)^\tau \cdot \exp(Q_j/T)}. \tag{5}$$

Here, $0 < \tau < 1$ limits the dominance of very large clusters, and the temperature T controls the influence of Q_k. The allocation sums to B_{total} and is measured in tokens.

Model-Conditioned Information Potential. RAP measures proxy adaptation speed through held-out loss reduction, an allocation signal related to $\mathcal{V}$-usable information (Xu et al., 2020; Ethayarajh et al., 2022). Each cluster probe is divided into disjoint adaptation and evaluation partitions, C_k^{adapt} and C_k^{eval}. From the shared reset initialization θ_{init}, we freeze the backbone and optimize the reset parameters for M steps on C_k^{adapt} to obtain θ_M. Both losses are measured on C_k^{eval}:

$$\Delta\mathcal{L}_k = \max\left\{0, \mathcal{L}(\theta_{init}; C_k^{eval}) - \mathcal{L}(\theta_M; C_k^{eval})\right\}. \tag{6}$$

Here $\mathcal{L}$ is the mean token-level loss. For all reported mixtures, $\Delta\mathcal{L}_k$ is the SmolLM-360M, $R = 0$ held-out loss reduction, shared by the 8B and 16B targets without proxy averaging. Large values encode rapid adaptation, and small values encode residual difficulty. We combine it with static quality as follows:

$$p_k = Q_k \exp\left(-\frac{\Delta\mathcal{L}_k}{\tau_{norm}}\right), \qquad r_k = 1 + \alpha p_k. \tag{7}$$

Here $\tau_{norm} = \max\{\mathbb{E}_k[\Delta\mathcal{L}_k], 10^{-8}\}$. The **information potential** $p_k \in [0, 1]$ is largest for high-quality clusters with small held-out loss reduction, and $r_k \in [1, 1 + \alpha]$. Appendix Figure 6 uses p_k as its color scale; final token shares follow Eq. 8.

Global Budget Normalization. We combine the two signals as $\mathcal{U}_k = n_k^{base} r_k$ and renormalize them to retain the total token budget:

$$n_k = \frac{B_{total}}{Z}\mathcal{U}_k, \qquad Z = \sum_{j=1}^{K} \mathcal{U}_j. \tag{8}$$

The normalization couples all cluster shares under the fixed budget (Xie et al., 2023a).

For clusters a and b, the final quota ratio is

$$\frac{n_a}{n_b} = \frac{n_a^{base}}{n_b^{base}} \frac{1 + \alpha p_a}{1 + \alpha p_b}. \tag{9}$$

The first factor contains mass, dispersion, and static quality; the second contains the proxy-conditioned correction. With $\alpha = 2$, the replay factor lies between 1 and 3 before global normalization, and the final quotas satisfy $\sum_k n_k = B_{total}$.

2.4 Intra-Cluster Selection

Given a token budget n_k for cluster C_k, we select examples using geometric diversity and length compensation.

Kernel-Based Diversity Sampling. We estimate the local density of a sample x with a Gaussian kernel over its nearest neighbors:

$$\rho(x) = \sum_{z \in \mathcal{N}_k(x)} \exp\left(-\frac{\|\mathbf{e}(x) - \mathbf{e}(z)\|^2}{2h^2}\right). \tag{10}$$

The base probability $P_{base}(x) \propto \rho(x)^{-1}$ down-weights dense regions and favors less common examples (Abbas et al., 2023; Tirumala et al., 2023; Zhang et al., 2025a; Lee et al., 2022).

Table 1: **Scaling comparison and multi-seed component ablation.** Panel A reports results for Random, DoReMi, CLIMB, UniGeM, and GRIP; Random, UniGeM, and GRIP use five paired seeds. Panel B reports means over three paired seeds under the 100B-token schedule; Avg. Score is shown as the mean ± standard deviation, and the task columns show seed means. Superscripts on highlighted rows give differences from UniGeM.

	Avg.	HumanEval		MBPP		LiveCodeBench	CruxEval		MultiPL-E
Model / Method	**Score**	**Pass@1**	**HE$^+$**	**Pass@1**	**MBPP$^+$**	**Pass@1**	**Input**	**Output**	**Avg.**
Panel A: Scaling Performance (300B tokens = 100B tokens × 3 epochs; 5 paired seeds)									
8B total / 1.4B active parameters									
Random 8B	28.4 ± 0.3	47.3	44.9	32.3	36.4	2.0	24.3	26.5	13.5
DoReMi 8B	32.7	52.0	49.1	36.6	42.3	5.6	30.2	27.8	18.0
CLIMB 8B	35.3	55.0	51.8	38.4	46.9	9.7	31.9	30.6	18.1
UniGeM 8B	36.5 ± 0.3	55.6	52.7	39.1	48.2	9.7	32.5	30.5	23.8
GRIP 8B	$37.4 \pm 0.4^{\uparrow 0.9}$	$58.8^{\uparrow 3.1}$	$51.6^{\downarrow 1.0}$	$39.4^{\uparrow 0.3}$	$49.4^{\uparrow 1.2}$	$10.6^{\uparrow 1.0}$	$33.1^{\uparrow 0.6}$	$31.2^{\uparrow 0.8}$	$25.1^{\uparrow 1.2}$
16B total / 1.4B active parameters									
Random 16B	32.2 ± 0.3	57.5	51.1	35.4	40.2	2.1	30.7	25.7	14.9
UniGeM 16B	39.6 ± 0.3	62.1	57.2	43.9	50.5	12.2	34.3	31.8	24.9
GRIP 16B	$40.9 \pm 0.4^{\uparrow 1.3}$	$65.9^{\uparrow 3.8}$	$56.9^{\downarrow 0.3}$	$42.7^{\downarrow 1.2}$	$51.5^{\uparrow 1.0}$	$15.9^{\uparrow 3.8}$	$34.2^{\downarrow 0.1}$	$31.5^{\downarrow 0.3}$	$28.5^{\uparrow 3.6}$
Panel B: Multi-Seed Component Ablation (GRIP-8B, 100B tokens; 3 paired seeds)									
Reference baselines									
1. Random	26.9 ± 0.3	45.6	44.4	30.2	34.1	2.3	22.1	26.0	10.3
UniGeM	34.3 ± 0.4	53.6	50.2	37.2	46.8	7.5	31.4	28.3	19.4
Incremental GRIP components									
2. + Static Budget	27.6 ± 0.2	47.4	45.7	31.5	34.3	2.2	22.7	25.4	11.6
3. + Static Replay	29.3 ± 0.3	49.0	47.1	33.0	37.5	3.2	24.4	26.5	13.8
4. + Loss Replay	31.5 ± 0.4	51.1	48.8	35.0	40.9	5.7	27.2	27.1	16.2
5. + Diversity	33.6 ± 0.5	53.8	50.2	36.7	44.4	7.1	28.9	28.1	19.7
6. GRIP (Full)	$35.5 \pm 0.3^{\uparrow 1.2}$	$55.4^{\uparrow 1.8}$	$48.5^{\downarrow 1.8}$	$36.0^{\downarrow 1.2}$	$48.9^{\uparrow 2.2}$	$9.7^{\uparrow 2.1}$	$32.6^{\uparrow 1.3}$	$27.7^{\downarrow 0.5}$	$25.1^{\uparrow 5.7}$

Length Correction. Because neighbor distance decreases with sequence length, inverse-density sampling can assign long sequences too little probability. We compensate with a length term:

$$P_{select}(x) \propto \underbrace{\frac{1}{\rho(x)}}_{\text{Diversity}} \cdot \underbrace{\left(\frac{\ell(x)}{\ell_{avg,k}}\right)^{\beta}}_{\text{Length Compensation}} . \tag{11}$$

Here, $\ell_{avg,k}$ is the average document length in cluster C_k, and $\beta \geq 0$ controls the correction strength; $\beta = 0$ recovers inverse-density sampling. Sampling proceeds without replacement. Near the quota boundary, we choose the highest-probability remaining document that fits the residual budget.

For two documents x and y in the same cluster, the normalized selection odds satisfy

$$\frac{P_{select}(x)}{P_{select}(y)} = \frac{\rho(y)}{\rho(x)} \left(\frac{\ell(x)}{\ell(y)}\right)^{\beta} . \tag{12}$$

At equal density, the reported $\beta = 0.3$ assigns a document three times longer a length factor of $3^{0.3} \approx 1.39$. Density remains the primary diversity signal; the sub-linear length factor offsets the measured compression of long-document embeddings.

3 Experimental Setup

Experiments use code pre-training, where heterogeneous programming languages coexist with exact syntax and long-range dependencies (Li et al., 2023; Feng et al., 2020; Guo et al., 2020).

3.1 Corpus and Model Configuration

Hybrid Corpus Composition. We construct a **100B-token candidate mixture** with a fixed CommonCrawl background and a selectable code component from The Stack v2 (Lozhkov et al., 2024).

GRIP embeds the code component with `Qwen/Qwen3-Embedding-8B` (Zhang et al., 2025c). The selected code is combined with the unchanged background data. The 100B-token setting uses one pass over this mixture; the **300B-token setting uses three epochs** over the same selected mixture.

Model Architecture. The target models are sparse Mixture-of-Experts Transformers with **8B total parameters and 32 experts** or **16B total parameters and 64 experts**. Both activate **1.4B parameters per token** (Liu et al., 2024). All runs start from scratch with the same architecture-specific recipe. The curation method is the experimental variable within each model scale and token budget.

Training Controls. Within each panel of Table 1, all methods share the candidate pool, CommonCrawl component, target architecture, optimizer, and token budget. Random, UniGeM, and the other baselines are trained from scratch. Curation changes the selectable code component. The 100B-token experiments use one pass over the selected mixture; the 300B-token schedule uses three epochs. Panel B fixes the 8B architecture and 100B-token budget and follows the sequence in Table 2. Appendix C lists the configuration.

3.2 GRIP Implementation Details

Probe Construction and Static Quality. The global probe budget is **0.5% of the candidate documents** and follows Eq. 4. `Qwen/Qwen3-235B-A22B` scores each probe document on the same four dimensions shown in Appendix Figure 5: *Code Quality*, *Algorithmic & Engineering Design*, *Training Suitability*, and *Knowledge Density* (Zheng et al., 2023; Qwen Team, 2025; Zhuang et al., 2025). For each cluster, we manually review several judge outputs against the source code and rubric, checking score-to-code alignment, consistency across dimensions, and adherence to the required schema. We average the judge scores within each cluster and rescale the result to $[0, 1]$ to obtain Q_k. Appendix A gives the rubric, output schema, and review protocol.

Rapid Adaptation Probe. Every reported mixture uses the allocation signal from **SmolLM-360M with** $R = 0$. We freeze the backbone, adapt the prediction head for **ten steps**, and measure $\Delta\mathcal{L}_k$ on a disjoint evaluation split. The sensitivity analysis uses SmolLM-135M, Qwen-2.5-0.5B Base/Coder, and $R \in \{1, 2, 4\}$ (Wang et al., 2025). Appendices B and C specify the protocol.

Metrics and Aggregation. HumanEval, HumanEval$^+$, MBPP, MBPP$^+$, and LiveCodeBench report Pass@1; CruxEval reports input/output accuracy; and MultiPL-E averages Pass@1 across languages. The Avg. Score is the unweighted mean of the eight metrics. Panel A aggregates five paired seeds for Random, UniGeM, and GRIP, while Panel B uses three. Appendix C gives the evaluation controls.

Paired Statistical Analysis. For each seed, we compute the unweighted mean of the eight task scores and form within-seed differences between methods. We report sample means and standard deviations across seeds, two-sided paired t tests, and 95% confidence intervals with $n - 1$ degrees of freedom. Holm correction is applied separately to the two model-scale comparisons and to the five ordered ablation transitions. The length-exponent sweep reports paired confidence intervals and unadjusted paired p values.

Hyperparameters. We set $M = 10$, $\alpha = 2.0$, $\tau = 0.5$, $T = 1.0$, and $\beta = 0.3$. The Diversity-to-GRIP transition changes only β from 0 to 0.3; Appendix E reports a three-seed sweep over $\{0, 0.1, 0.3, 0.5\}$.

3.3 Evaluation Protocol

Baselines and Ablations. Panel A compares GRIP with Random, DoReMi, CLIMB, and UniGeM, spanning uniform sampling, domain reweighting, model-guided mixing, and hierarchical geometric selection (Xie et al., 2023a; Diao et al., 2025; Wang et al., 2026). Panel B follows the ordered sequence Random, Static Budget, Static Replay, Loss Replay, Diversity, and GRIP. Table 2 states the allocation and selection rules for each variant, and Appendix D gives their implementation definitions.

Table 2: Definition of the ordered 100B-token ablation variants. The two rule columns state each row's configuration; adjacent rows isolate the corresponding allocation or selection change.

Variant	Inter-cluster budget	Intra-cluster selection
Random	Uniform token share	Uniform
Static Budget	Mass, dispersion, Q_k	Uniform
Static Replay	Baseline + quality replay	Uniform
Loss Replay	Baseline + RAP replay	Uniform
Diversity	Baseline + RAP replay	$\rho(x)^{-1}$ ($\beta = 0$)
GRIP	Baseline + RAP replay	$\rho(x)^{-1}(\ell/\ell_{avg})^{0.3}$

Benchmarks. We report the following benchmark groups:

- **Code generation:** HumanEval and MBPP (Chen et al., 2021; Austin et al., 2021), including their EvalPlus variants (Liu et al., 2023).
- **Recent and execution-based tasks:** LiveCodeBench (Jain et al., 2024) and CruxEval (Gu et al., 2024).
- **Multilingual generation:** MultiPL-E (Cassano et al., 2023).

4 Results

4.1 Scaling Results

Table 1 reports results for the 8B and 16B models. Under the 300B-token schedule, Panel A compares GRIP with random sampling, DoReMi for domain reweighting, CLIMB for model-guided mixing, and UniGeM for hierarchical geometric selection (Xie et al., 2023a; Diao et al., 2025; Wang et al., 2026).

Average performance. GRIP exceeds Random by **9.0 points** at 8B and **8.7 points** at 16B. Against UniGeM, five paired seeds give gains of **+0.90 points** (95% CI: 0.34–1.45; Holm-adjusted $p = 0.011$) and **+1.30 points** (0.78–1.81; Holm-adjusted $p = 0.004$), respectively.

Baseline comparison. GRIP-8B reaches 37.4 versus 32.7 for DoReMi (Xie et al., 2023a), 35.3 for CLIMB (Diao et al., 2025), and 36.5 for UniGeM. Its largest margins over CLIMB occur on MultiPL-E, HumanEval, and MBPP$^+$.

Task pattern. HumanEval and MultiPL-E improve in all five pairs at both scales. The largest 16B gains are **+3.8** on HumanEval, **+3.8** on LiveCodeBench, and **+3.6** on MultiPL-E. Appendix E gives all task-level differences.

Seed-level results. At 300B tokens, LiveCodeBench improves in four of five 8B pairs and all five 16B pairs; MBPP$^+$ improves in four pairs at each scale, with one 16B tie. In the three-seed 100B comparison with UniGeM, GRIP improves HumanEval, MBPP$^+$, LiveCodeBench, CruxEval Input, and MultiPL-E in every pair.

Token trajectory. Figure 3(a) follows three matched runs from the 300B-token schedule. The GRIP–Random gap is **8.6 points at 100B** and 9.0 at 300B for 8B models; the corresponding 16B gaps are 8.2 and 8.7. In the separate compute-matched 100B setting, GRIP is **1.2 points** above UniGeM.

4.2 Ordered Component Ablation

Panel B of Table 1 reports the ordered GRIP variants under the 100B-token schedule; Table 2 identifies the rule changed at each transition.

- **Static allocation:** Static Budget raises the average score from 26.9 to 27.6 (**+0.7 points**). Static Replay increases it to 29.3 (**+1.7 points**).
- **Adaptation-based replay:** Replacing quality-based replay with Loss Replay changes the average from 29.3 to 31.5 (**+2.2 points**).

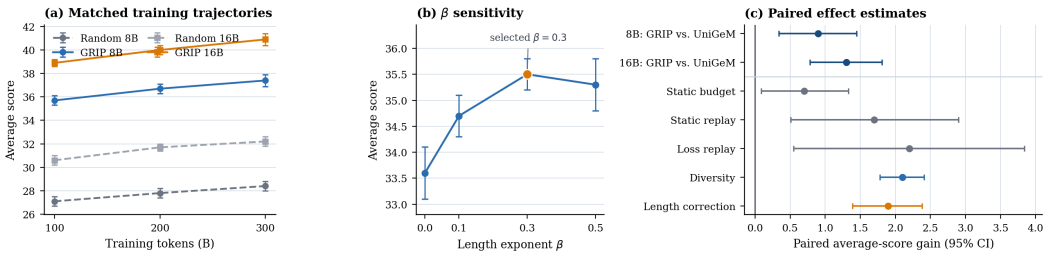


Figure 3: **Multi-seed results and sensitivity.** (a) Matched training trajectories. (b) Length-exponent sweep. (c) Paired gains with 95% confidence intervals for the 300B-token comparisons and ordered 100B-token ablations.

- **Geometric diversity:** Inverse-density sampling increases the average from 31.5 to 33.6 (**+2.1 points**). MultiPL-E rises from 16.2 to 19.7.
- **Length correction:** GRIP changes the average from 33.6 to 35.5 (**+1.9 points**; paired 95% CI: 1.39–2.38). The largest task gains are on MultiPL-E (19.7 to 25.1, **+5.4 points**), MBPP$^+$ (44.4 to 48.9, **+4.6 points**), and CruxEval Input (28.9 to 32.6, **+3.8 points**).

Appendix E reports the task-level profile, paired intervals, ordered-test correction, and β sensitivity.

Paired inference. Figure 3(c) reports paired 95% CIs of 0.09–1.33, 0.51–2.90, 0.55–3.84, 1.78–2.41, and 1.39–2.38 points for the five ordered increments. Holm-adjusted p values are 0.0059 for Diversity and 0.0149 for Length Correction.

Length sensitivity. Figure 3(b) plots the three-seed averages for $\beta \in \{0, 0.1, 0.3, 0.5\}$: 33.6, 34.7, 35.5, and 35.3. The highest mean occurs at $\beta = 0.3$, with a 1.89-point difference from $\beta = 0$ (95% CI: 1.39–2.38; $p = 0.0037$).

Task-level component profile. The ordered transitions affect distinct benchmark groups. Static Budget produces its largest gains on HumanEval (**+1.8**), MBPP (**+1.3**), and MultiPL-E (**+1.3**). Static Replay adds **+3.2** on MBPP$^+$ and **+2.2** on MultiPL-E. Replacing static replay with Loss Replay adds **+3.4** on MBPP$^+$, **+2.8** on CruxEval Input, **+2.5** on LiveCodeBench, and **+2.4** on MultiPL-E. Inverse-density sampling improves all eight reported tasks, led by MultiPL-E (**+3.5**), MBPP$^+$ (**+3.4**), and HumanEval (**+2.7**). Length correction then adds **+5.4** on MultiPL-E, **+4.6** on MBPP$^+$, **+3.8** on CruxEval Input, and **+2.6** on LiveCodeBench. These task profiles separate the effects of cluster allocation, geometric coverage, and document length.

4.3 Allocation Signals and Length Geometry

Static Quality and Loss Dynamics. Static quality Q_k and adaptation delta $\Delta\mathcal{L}_k$ have a **weak negative linear correlation** ($r \approx -0.202$) in Figure 4(a). Clusters with similar quality scores have different adaptation deltas. Appendix A contains the quality rubric, and Appendix Figure 5 reports the four judge-score distributions (Zhuang et al., 2025; Wang et al., 2025).

Cross-Proxy Comparison. The reported allocation uses **SmolLM-360M with** $R = 0$; SmolLM-135M and Qwen-2.5-0.5B Base/Coder across $R \in \{0, 1, 2, 4\}$ provide sensitivity comparisons. Figure 4(b) reports initial loss, adaptation delta, and information potential for two representative proxies. RAP accounts for **less than 1%** of the target-model training cost; Appendices B and C report the settings.

The 8B and 16B target models share the same SmolLM-360M-derived $\Delta\mathcal{L}_k$ values, cluster allocation, and selected mixture. The additional proxy configurations measure sensitivity across architecture and reset depth.

Using the same allocation at both target scales keeps the data mixture fixed while model capacity changes. The repeated paired gains therefore come from training the two target architectures on the

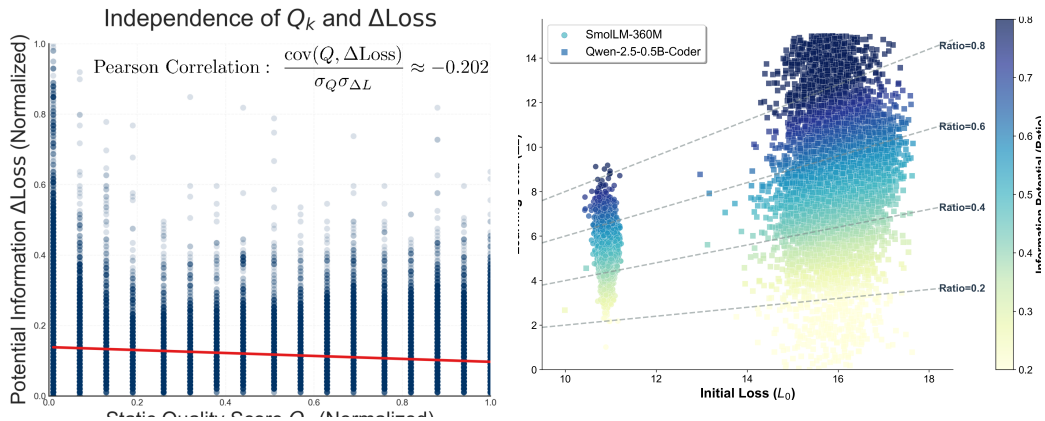


Figure 4: **Allocation-signal diagnostics.** (a) Static quality Q_k and adaptation delta $\Delta\mathcal{L}_k$ have a weak negative correlation ($r \approx -0.202$). (b) Initial loss, adaptation delta, and information potential p_k for SmolLM-360M and Qwen-2.5-0.5B-Coder.

same selected corpus rather than retuning the allocation for each scale. Their shared task pattern also matches the component ablation: allocation affects broad code-generation performance, while geometric and length-aware selection contribute strongly on multilingual and execution-based tasks.

Geometric Collapse and Length Rectification. After 32K chunking and document-level aggregation, normalized k-NN distance falls below **0.2** for documents longer than **50k tokens** (Figure 2). Inverse-density sampling assigns these rare long documents low probability. Length correction raises MultiPL-E from **19.7 to 25.1** in the component ablation. Appendix F reports the residual length analysis.

Preprocessing Cost and Mixture Manifest. RAP uses **less than 1%** of the target-model training cost. Corpus embedding, clustering, and quality scoring run once before training. The mixture manifest stores document identifiers, cluster assignments, selection weights, sampling seeds, assigned token counts, and realized token counts. Appendix C lists the configuration of each stage.

Allocation and Selection Trace. Each cluster record contains its mass, dispersion, quality score, adaptation delta, baseline quota, replay multiplier, and final quota. Each selected document records its cluster, neighbor density, length factor, sampling weight, and realized token contribution. The two record levels separate the reason a cluster receives capacity from the reason a document fills that capacity. Summed final quotas match the requested budget up to integer rounding, and realized tokens remain within their assigned cluster quotas.

5 Conclusion

GRIP combines model-conditioned allocation with length-aware geometric selection. The ordered 100B-token ablation assigns **2.2 average-score points** to adaptation-based replay, **2.1** to inverse-density diversity, and **1.9** to length correction. The task-level differences show that allocation changes the semantic mixture, while geometric and length-aware sampling determine the coverage within each allocated region. Across five paired seeds at 300B tokens, GRIP exceeds UniGeM by **0.9 points** at 8B (95% CI: 0.34–1.45) and **1.3 points** at 16B (0.78–1.81). The GRIP–Random margin is already 8.6 points at the 100B checkpoint for the 8B model and reaches 9.0 points at 300B; the corresponding 16B margins are 8.2 and 8.7. HumanEval and MultiPL-E improve in every paired run at both model scales, with additional recurring gains on LiveCodeBench and MBPP$^+$.

AI Use Statement

Generative AI was used for language editing and LaTeX formatting.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

Alon Albalak et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable k-means clustering via lightweight coresets. In *Proceedings of KDD*, 2018.

Zoltán Botta-Dukát. Rao's quadratic entropy as a measure of functional diversity based on multiple traits. *Journal of vegetation science*, 16(5):533–540, 2005.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, et al. Multipl-e: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 49(7):3675–3691, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*, 2019.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with v-usable information. In *International Conference on Machine Learning*, pages 5988–6008. PMLR, 2022.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. Codebert: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1536–1547, 2020.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, et al. Language models scale reliably with over-training and on downstream tasks. *arXiv preprint arXiv:2403.08540*, 2024.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22702–22711, 2024.

Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. Cruxeval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*, 2024. URL https://arxiv.org/abs/2401.03065.

Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie Liu, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, et al. Graphcodebert: Pre-training code representations with data flow. *arXiv preprint arXiv:2009.08366*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *Advances in Neural Information Processing Systems*, 35:30016–30030, 2022.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021. doi: 10.1109/TBDATA.2019.2921572.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of ACL*, 2022.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Fengze Liu, Weidong Zhou, Binbin Liu, Zhimiao Yu, Yifan Zhang, Haobin Lin, Yifeng Yu, Bingni Zhang, Xiaohuan Zhou, Taifeng Wang, et al. Quadmix: Quality-diversity balanced data selection for efficient llm pretraining. *arXiv preprint arXiv:2504.16511*, 2025.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *arXiv preprint arXiv:2305.01210*, 2023.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*, 2024.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S Morcos. D4: Improving llm pretraining via document de-duplication and diversification. In *NeurIPS*, 2023.

Changhao Wang, Yunfei Yu, Xinhao Yao, Jiaolong Yang, Lu Yu, Junpeng Fang, Riccardo Cantoro, Chaobo Li, Qing Cui, and Jun Zhou. UniGeM: Unifying data mixing and selection via geometric exploration and mining. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 20696–20714, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1037. URL https://aclanthology.org/2026.findings-acl.1037/.

Jiachen T Wang, Tong Wu, Kaifeng Lyu, James Zou, Dawn Song, Ruoxi Jia, and Prateek Mittal. Can small training runs reliably guide data curation? rethinking proxy-model practice. *arXiv preprint arXiv:2512.24503*, 2025.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *NeurIPS*, 2023a.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023b.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*, 2020.

Zichun Yu, Spandan Das, and Chenyan Xiong. Mates: Model-aware data selection for efficient pretraining with data influence models. *Advances in Neural Information Processing Systems*, 37: 108735–108759, 2024.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Qiu Jiantao, Lei Cao, Ju Fan, Ye Yuan, Guoren Wang, and Conghui He. Harnessing diversity for important data selection in pretraining large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=bMC1t7eLRc.

Jia Zhang, Chen-Xi Zhang, Yao Liu, Yi-Xuan Jin, Xiao-Wen Yang, Bo Zheng, Yi Liu, and Lan-Zhe Guo. D3: Diversity, difficulty, and dependability-aware data selection for sample-efficient llm instruction tuning. *arXiv preprint arXiv:2503.11441*, 2025b.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025c.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yanping Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623, 2023.

Yuqi Zhou, Sunhao Dai, Zhanshuo Cao, Xiao Zhang, and Jun Xu. Length-induced embedding collapse in plm-based models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28767–28791, 2025.

Xinlin Zhuang, Jiahui Peng, Ren Ma, Yinfan Wang, Tianyi Bai, Xingjian Wei, Jiantao Qiu, Chi Zhang, Ying Qian, and Conghui He. Meta-rater: A multi-dimensional data selection method for pre-training language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10856–10896, July 2025.

A Prompt for Static Quality Assessment

To assess code samples for pre-training, we use `Qwen/Qwen3-235B-A22B` with an LLM-as-a-judge prompt covering syntax, engineering design, educational value, and technical depth. The instruction is reproduced below.

System Instruction: Comprehensive Code Audit Protocol
You are acting as a **Senior Technical Auditor**. Your objective is to rigorously evaluate the provided code snippet against four core performance indicators. Unlike simple correctness checks, you must assess the code's readiness for high-quality pre-training datasets. Use a granular **0 to 10 scale** for each dimension.

1. Code Quality (0–10)
Focus: Syntax, naming, and readability.

- **0–3 (Low):** Severe syntax errors; code cannot run; messy formatting or naming makes it unreadable.
- **4–7 (Mid):** Executable and generally correct; minor formatting issues or inconsistent naming; readable but not polished.
- **8–10 (High):** Error-free and standard compliant (e.g., PEP 8); precise naming; clean and professional structure.

2. Algorithmic & Engineering Design (0–10)

Focus: Modularity, robustness, and structure.

- **0–3 (Low):** All code is in one block; there are no functions or classes; input validation and error handling are absent.
- **4–7 (Mid):** The code is split into functions and includes basic error checking, but has limited extensibility.
- **8–10 (High):** The code is highly modular, includes robust error handling, and follows strong engineering practices.

3. Training Suitability (0–10)

Focus: Educational value and clarity for learning.

- **0–3 (Low):** No comments; confusing logic; unexplained magic numbers; difficult for a model or human to learn from.
- **4–7 (Mid):** Clear logic with basic comments for key steps; suitable for general training.
- **8–10 (High):** Comments explain the rationale; the code demonstrates best practices and is highly suitable for teaching.

4. Knowledge Density (0–10)

Focus: Technical insight and complexity.

- **0–3 (Low):** Trivial code, such as simple print statements or basic loops, with little domain knowledge.
- **4–7 (Mid):** Standard algorithms or typical business logic; useful but not distinctive.
- **8–10 (High):** Complex algorithms, deep optimizations, or cross-domain knowledge with high technical value.

Response Requirement: Analyze the input code and output a JSON object strictly adhering to this schema:

```
{
  "code_quality_score": <int 0--10>,
  "algorithmic_engineering_score": <int 0--10>,
  "training_suitability_score": <int 0--10>,
  "knowledge_density_score": <int 0--10>
}
```

Target Code Snippet: `$content`

A.1 Manual Review of Judge Outputs

For each cluster, we manually review several quality-judge outputs before computing the cluster score. The review compares the source code, the four scores, and the rubric, then checks score–property alignment, consistency across dimensions, JSON validity, and the 0–10 range. We compute Q_k from the reviewed cluster scores.

B End-to-End Selection Procedure

The offline curation pipeline runs once before target-model pre-training:

1. Split each document into contiguous chunks of at most 32,768 tokens. Embed each chunk with the EOS-token representation from `Qwen/Qwen3-Embedding-8B`, combine chunk embeddings by a token-weighted mean, normalize the document representation, and partition the corpus with spherical k-means.
2. Compute cluster size and geometric dispersion, then allocate the 0.5% probe according to the dispersion-aware probe rule in the main paper.
3. Score probe documents along code quality, algorithmic & engineering design, training suitability, and knowledge density. For each cluster, manually review several judge outputs against the source code, rubric, and required schema, then average and rescale the cluster's judge scores to obtain $Q_k \in [0, 1]$.
4. Split each cluster probe into disjoint partitions C_k^{adapt} and C_k^{eval}. For the final allocation, restore SmolLM-360M with $R = 0$, freeze its backbone, and optimize the prediction head for ten steps on C_k^{adapt}. Compute $\Delta\mathcal{L}_k$ from the pre- and post-adaptation losses evaluated on the same held-

out partition C_k^{eval}. SmolLM-135M, Qwen-2.5-0.5B Base/Coder, and $R \in \{1, 2, 4\}$ form the sensitivity analysis.

5. Combine cluster mass, dispersion, and Q_k to obtain the baseline token budget. Set $\tau_{norm} = \max\{\mathbb{E}_k[\Delta\mathcal{L}_k], 10^{-8}\}$, compute $p_k = Q_k \exp(-\Delta\mathcal{L}_k/\tau_{norm})$ and $r_k = 1 + \alpha p_k$, and renormalize all cluster utilities to the global token constraint.
6. Estimate local kernel density inside each cluster and sample without replacement using the length-rectified inverse-density probability. Stop before exceeding the cluster quota, then use the highest-probability remaining document that fits the residual budget.
7. Merge the selected code component with the unchanged CommonCrawl background. The 100B-token schedule uses one pass; the 300B-token schedule reuses the same selected mixture for three epochs.

All allocation and selection statistics are computed before target-model training. Benchmark decontamination is a separate preprocessing stage.

B.1 Signal Interpretation and Budget Invariants

The baseline budget uses corpus mass, dispersion, and static quality. RAP modifies that budget using proxy behavior. Because

$$p_k = Q_k \exp(-\Delta\mathcal{L}_k/\tau_{norm}), \tag{13}$$

p_k is largest for clusters with high static quality and small held-out loss reduction. Rapid adaptation produces a smaller replay adjustment; high quality combined with slow adaptation produces a larger adjustment. Negative loss reductions are clipped to zero.

The baseline quotas satisfy $\sum_k n_k^{base} = B_{total}$. After applying $r_k = 1 + \alpha p_k$, global normalization gives $\sum_k n_k = B_{total}$. Replay redistributes a fixed budget. From $p_k \in [0, 1]$, the pre-normalization replay range is $r_k \in [1, 1 + \alpha]$.

The within-cluster sampler preserves n_k. Density and length determine document order inside C_k. Setting $\beta = 0$ gives the Diversity variant; replacing inverse-density sampling with uniform sampling gives Loss Replay. The substitutions define the adjacent ablation rows.

B.2 Detailed Mathematical Derivation

The derivation below distinguishes document counts from token budgets and writes each normalization explicitly.

Notation and units. For cluster C_k, $N_k = |C_k|$ is the number of candidate documents, σ_k is dimensionless embedding dispersion, and $Q_k \in [0, 1]$ is the rescaled mean quality score. The global probe size B_{probe} is measured in documents; the training budget B_{total}, baseline quota n_k^{base}, and final quota n_k are measured in tokens. For document x, $\ell(x)$ is its token length and $\rho(x)$ is its local kernel-density score.

Document representation and dispersion. Document x is divided into chunks $x_1, \ldots, x_m$ with lengths $\ell_i = \ell(x_i)$ and normalized chunk embeddings $\mathbf{e}_i$. Before the final normalization, its aggregate representation is

$$\widetilde{\mathbf{e}}(x) = \frac{\sum_{i=1}^m \ell_i \mathbf{e}_i}{\sum_{i=1}^m \ell_i}, \qquad \mathbf{e}(x) = \frac{\widetilde{\mathbf{e}}(x)}{\|\widetilde{\mathbf{e}}(x)\|_2}. \tag{14}$$

Chunk i has weight $\ell_i / \sum_j \ell_j$; longer chunks contribute proportionally more, and every encoder input remains within 32,768 tokens. After spherical k-means, the root-mean-square dispersion is

$$\sigma_k^2 = \frac{1}{N_k} \sum_{x \in C_k} \|\mathbf{e}(x) - \mu_k\|_2^2. \tag{15}$$

Squaring makes deviations nonnegative, averaging removes direct dependence on N_k, and the square root returns the result to the distance scale.

Normalized probe allocation. The normalized probe rule is

$$n_k^{probe} = B_{probe} \frac{N_k \sigma_k}{\sum_{j=1}^{K} N_j \sigma_j}. \tag{16}$$

Summing Eq. 16 over k gives $\sum_k n_k^{probe} = B_{probe}$. The relative probe size of clusters a and b is

$$\frac{n_a^{probe}}{n_b^{probe}} = \frac{N_a \sigma_a}{N_b \sigma_b}. \tag{17}$$

Doubling mass or dispersion doubles the unrounded probe allocation. Integer rounding and residual assignment preserve the total probe size.

Static budget as a normalized exponential score. Define the positive unnormalized baseline weight

$$w_k = (N_k \sigma_k)^\tau \exp(Q_k / T). \tag{18}$$

Then the baseline token share and quota are

$$b_k = \frac{w_k}{\sum_j w_j}, \qquad n_k^{base} = B_{total} b_k. \tag{19}$$

Because $\sum_k b_k = 1$, Eq. 19 gives $\sum_k n_k^{base} = B_{total}$. In log space, the three static terms are

$$\log w_k = \tau \log N_k + \tau \log \sigma_k + Q_k / T. \tag{20}$$

For two clusters a and b, their baseline quota ratio is

$$\frac{n_a^{base}}{n_b^{base}} = \left(\frac{N_a \sigma_a}{N_b \sigma_b} \right)^\tau \exp\left(\frac{Q_a - Q_b}{T} \right). \tag{21}$$

At $\tau = 0.5$, a fourfold change in $N_k \sigma_k$ produces a twofold change before quality weighting. At $T = 1$, a quality difference of 0.2 contributes a factor $e^{0.2} \approx 1.22$. Global normalization couples the cluster quotas.

Held-out adaptation signal. Let $C_k^{adapt} \cap C_k^{eval} = \varnothing$. For the reported allocation, the proxy is SmolLM-360M with $R = 0$. Starting from θ_{init}, its prediction head is adapted for M steps using only C_k^{adapt}, producing θ_M. Define $\mathcal{L}_k^{init} = \mathcal{L}(\theta_{init}; C_k^{eval})$ and $\mathcal{L}_k^M = \mathcal{L}(\theta_M; C_k^{eval})$ as the mean token losses on the same held-out evaluation partition. The nonnegative reduction is

$$\Delta \mathcal{L}_k = \max\{0, \mathcal{L}_k^{init} - \mathcal{L}_k^M\}. \tag{22}$$

The 8B and 16B target mixtures share the SmolLM-360M evaluation-partition loss reduction clipped at zero. The scale

$$\tau_{norm} = \max\left\{ K^{-1} \sum_{k=1}^{K} \Delta \mathcal{L}_k, 10^{-8} \right\} \tag{23}$$

normalizes loss reduction by its across-cluster mean; the floor handles a zero mean.

The information potential and replay multiplier are

$$p_k = Q_k e^{-\Delta \mathcal{L}_k / \tau_{norm}}, \qquad r_k = 1 + \alpha p_k. \tag{24}$$

The constraints $Q_k \in [0, 1]$, $\Delta \mathcal{L}_k \geq 0$, and $e^{-u} \in (0, 1]$ for $u \geq 0$ give $0 \leq p_k \leq 1$ and

$$1 \leq r_k \leq 1 + \alpha. \tag{25}$$

For the reported $\alpha = 2$, the pre-normalization replay multiplier lies in $[1, 3]$. Its local sensitivities are

$$\frac{\partial p_k}{\partial Q_k} = e^{-\Delta \mathcal{L}_k / \tau_{norm}} \geq 0, \qquad \frac{\partial p_k}{\partial \Delta \mathcal{L}_k} = -\frac{p_k}{\tau_{norm}} \leq 0. \tag{26}$$

The derivatives are positive in quality and negative in adaptation reduction, assigning larger replay factors to high-quality, slowly adapting clusters.

Table 3: Final-allocation configuration and the accompanying proxy sensitivity settings.

Component	Setting	Component	Setting
Candidate mixture	100B tokens	300B schedule	Three passes over selected mixture
Embedding model	`Qwen/Qwen3-Embedding-8B`	Quality judge	`Qwen/Qwen3-235B-A22B`
Embedding window	32,768 tokens	Embedding pooling	Final EOS-token state
Long documents	Contiguous chunks	Chunk aggregation	Token-weighted mean, then normalize
Global probe budget	0.5% of documents	Quality range	Rescaled to $[0, 1]$
Final allocation proxy	SmolLM-360M, $R = 0$	Sensitivity proxies	SmolLM-135M; Qwen-2.5-0.5B Base/Coder
Target-scale mapping	Shared by 8B and 16B	Sensitivity depths	$\{0, 1, 2, 4\}$
Adaptation steps M	10	Probe optimizer	AdamW, 3×10^{-4}
RAP evaluation	Held-out probe partition	Potential	$p_k = Q_k e^{-\Delta\mathcal{L}_k/\tau_{norm}}$
Potential scale	$\tau_{norm} = \max\{\mathbb{E}_k[\Delta\mathcal{L}_k], 10^{-8}\}$	Replay range	$r_k \in [1, 1 + \alpha]$
Replay strength α	2.0	Budget exponent τ	0.5
Quality temperature T	1.0	Length exponent β	0.3
No-length ablation	$\beta = 0$	Training hardware	8× NVIDIA H800
Target models	8B / 16B MoE	Active parameters	1.4B per token

Final global normalization and conservation. Combining the static share with replay gives unnormalized utility

$$\mathcal{U}_k = n_k^{base} r_k = B_{total} b_k r_k. \tag{27}$$

The final quota is

$$n_k = B_{total} \frac{b_k r_k}{\sum_j b_j r_j}. \tag{28}$$

Summing Eq. 28 proves budget conservation:

$$\sum_k n_k = B_{total} \frac{\sum_k b_k r_k}{\sum_j b_j r_j} = B_{total}. \tag{29}$$

Replay redistributes a fixed token budget. Pairwise final odds are

$$\frac{n_a}{n_b} = \frac{n_a^{base}}{n_b^{base}} \frac{1 + \alpha p_a}{1 + \alpha p_b}. \tag{30}$$

The first factor is the static preference and the second is the model-conditioned correction.

Normalized intra-cluster sampling. For $x \in C_k$, define

$$q_k(x) = \rho(x)^{-1} \left(\frac{\ell(x)}{\ell_{avg,k}} \right)^{\beta}, \qquad P_k(x) = \frac{q_k(x)}{\sum_{y \in C_k} q_k(y)}. \tag{31}$$

The denominator normalizes P_k over C_k. For documents $x, y \in C_k$,

$$\frac{P_k(x)}{P_k(y)} = \frac{\rho(y)}{\rho(x)} \left(\frac{\ell(x)}{\ell(y)} \right)^{\beta}. \tag{32}$$

The cluster-average length cancels from the ratio. For equal-density documents with a length ratio of three, $\beta = 0.3$ contributes a factor $3^{0.3} \approx 1.39$. At $\beta = 0$, Eq. 32 reduces to inverse-density sampling. Sampling proceeds without replacement, with renormalization after each draw. The final highest-probability document that fits the residual budget reduces unused slack without exceeding n_k.

C Implementation and Training Configuration

Within each panel, methods share the target architecture, optimizer, token budget, background corpus, evaluation harness, and decoding configuration. The curation method changes the selectable code component. Random, UniGeM, and GRIP use five paired seeds at each target scale for the 300B-token comparison; the 100B-token ablation uses three paired seeds.

Statistical protocol. For each seed, we first compute the unweighted mean of the eight task scores, then form within-pair differences between methods. We report the mean and sample standard deviation across seeds, two-sided paired t tests, and 95% confidence intervals using $n - 1$ degrees of freedom. Holm correction is applied separately to the two principal scale comparisons and to the five ordered ablation transitions. The β sweep reports paired confidence intervals and unadjusted paired p values.

D Baseline and Ablation Definitions

Random
: Uniform sampling from the selectable code component.

Static Budget
: Cluster quotas use corpus mass, dispersion, and static quality; documents are sampled uniformly within clusters.

Static Replay
: Adds replay weights computed from static quality in place of proxy adaptation loss.

Loss Replay
: Replaces quality-based replay with $p_k = Q_k \exp(-\Delta\mathcal{L}_k/\tau_{norm})$ and $r_k = 1 + \alpha p_k$, while retaining uniform within-cluster sampling.

Diversity
: Adds inverse-density sampling but disables length correction by setting $\beta = 0$.

GRIP
: Combines static allocation, adaptation-based replay, inverse-density sampling, and length correction.

Each adjacent pair changes one recorded component from Random sampling to GRIP.

E Scaling, Ablation, and Sensitivity Results

E.1 Quality and Proxy Diagnostics

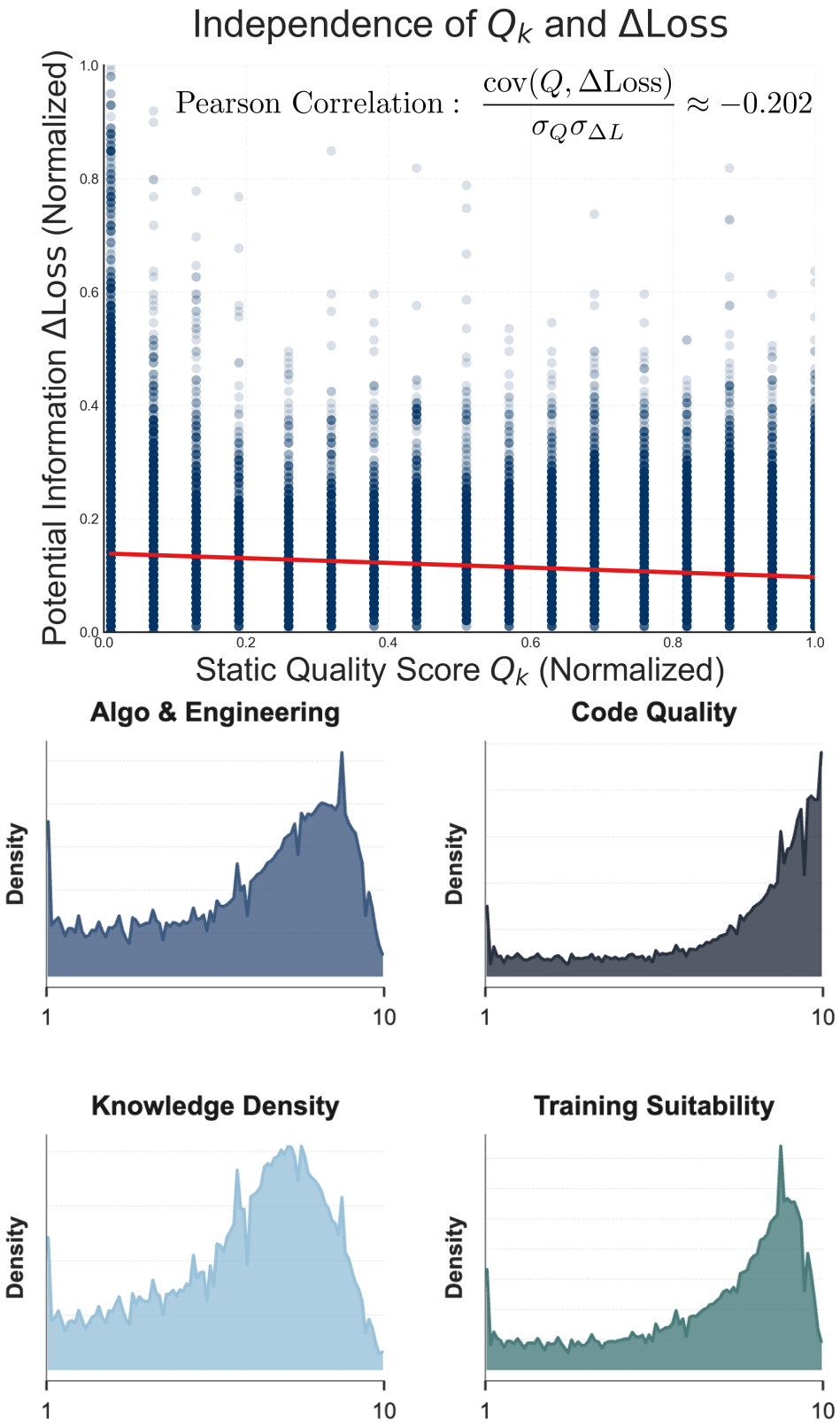


Figure 5: Static quality and adaptation dynamics. Cluster-level quality and normalized adaptation delta have a weak negative Pearson correlation ($r \approx -0.202$). The lower panels use the four quality dimensions from Appendix A.

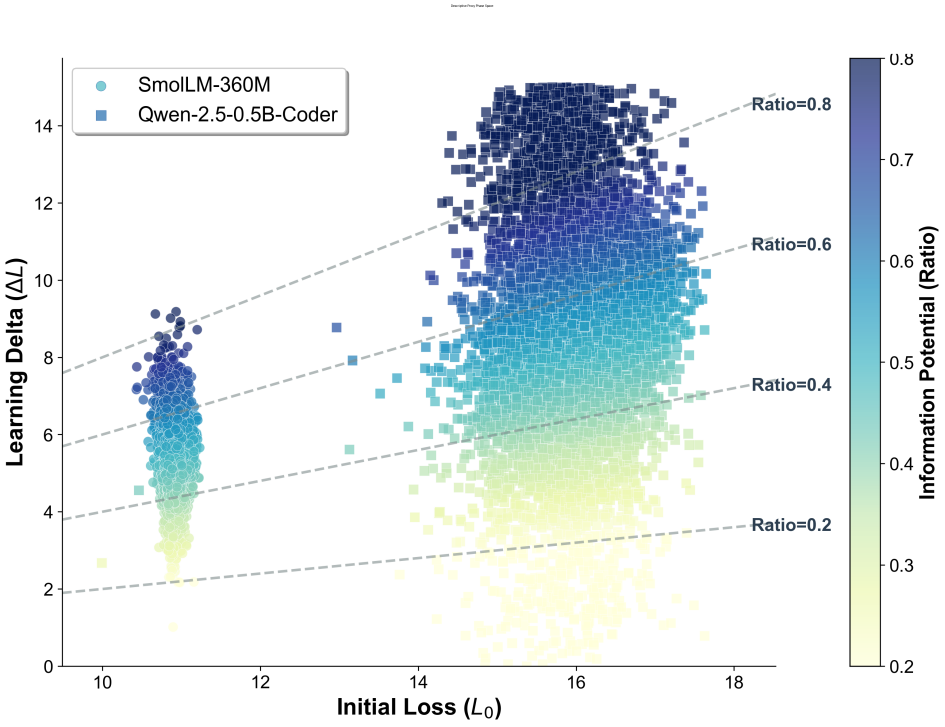


Figure 6: Cross-proxy comparison of initial loss, adaptation delta, and information potential p_k for two representative proxies.

E.2 Scaling Results at 300B Tokens

Table 4 lists task-level results. Random, UniGeM, and GRIP aggregate five paired seeds.

Eight-billion-parameter comparison. GRIP-8B raises the average from **36.5 to 37.4** relative to UniGeM, a mean paired difference of **+0.895 points** (95% CI: 0.337–1.453; paired $t(4) = 4.46$, two-sided $p = 0.011$). The largest task differences are HumanEval (**+3.14**), MultiPL-E (**+1.24**), and MBPP$^+$ (**+1.20**); HumanEval and MultiPL-E improve in all five pairs. The average margin over Random 8B is **8.995 points**.

Sixteen-billion-parameter comparison. GRIP-16B raises the average from **39.6 to 40.9**, a mean paired difference of **+1.298 points** (95% CI: 0.781–1.814; paired $t(4) = 6.97$, two-sided $p = 0.0022$). The largest task differences are HumanEval (**+3.80**), LiveCodeBench (**+3.78**), and MultiPL-E (**+3.60**), all positive in five pairs. The average margin over Random 16B is **8.690 points**. Holm-adjusted p values for the 8B and 16B GRIP–UniGeM tests are 0.011 and 0.0044.

Average definition. The displayed average is the unweighted arithmetic mean of the eight task columns. For example, the GRIP-8B value is

$$\begin{aligned} \mathrm{Avg}_{\text{GRIP-8B}} &= \frac{1}{8}\big(58.8 + 51.6 + 39.4 + 49.4 \\ &\qquad + 10.6 + 33.1 + 31.2 + 25.1\big) \\ &= 37.4. \end{aligned} \tag{33}$$

E.3 Absolute Scores

Table 5 reports the absolute seed means underlying the adjacent differences. Standard deviations are available for the average score because that is the aggregate reported across the three paired runs; task columns report seed means, matching Panel B of the main table.

Absolute scores. The ordered averages are **26.9** for Random, **27.6** for Static Budget, **29.3** for Static Replay, **31.5** for Loss Replay, **33.6** for Diversity, and **35.5** for GRIP. GRIP is **1.2 points above UniGeM** at 100B tokens and has the highest displayed score on HumanEval, MBPP$^+$, LiveCodeBench, CruxEval Input, and MultiPL-E.

Diversity raises HumanEval$^+$ and MBPP relative to Loss Replay. Length correction contributes its largest gains on MultiPL-E, MBPP$^+$, and CruxEval Input. Across both stages, the average rises from 31.5 to 35.5.

E.4 Adjacent Component Changes

Paired inference. Each task profile is the mean over three paired seeds. Paired 95% confidence intervals for the five successive average-score changes are 0.09–1.33, 0.51–2.90, 0.55–3.84, 1.78–2.41, and 1.39–2.38 points. Holm-adjusted p values are 0.0059 for Diversity and 0.0149 for Length Correction.

E.5 Length-Exponent Sensitivity

Table 7 reports the three-seed length-exponent sweep. Paired differences compare each positive value with inverse-density sampling at $\beta = 0$.

Table 4: Task-level scaling results under the 300B-token schedule. Random, UniGeM, and GRIP report means over five paired seeds; Avg. includes the seed standard deviation.

Model / Method	Avg.	HE	HE$^+$	MBPP	MBPP$^+$	LCB	Crux-I	Crux-O	MultiPL-E
Random 8B	28.4 ± 0.3	47.3	44.9	32.3	36.4	2.0	24.3	26.5	13.5
DoReMi 8B	32.7	52.0	49.1	36.6	42.3	5.6	30.2	27.8	18.0
CLIMB 8B	35.3	55.0	51.8	38.4	46.9	9.7	31.9	30.6	18.1
UniGeM 8B	36.5 ± 0.3	55.6	**52.7**	39.1	48.2	9.7	32.5	30.5	23.8
GRIP 8B	$\mathbf{37.4 \pm 0.4}$	**58.8**	51.6	**39.4**	**49.4**	**10.6**	**33.1**	**31.2**	**25.1**
Random 16B	32.2 ± 0.3	57.5	51.1	35.4	40.2	2.1	30.7	25.7	14.9
UniGeM 16B	39.6 ± 0.3	62.1	**57.2**	**43.9**	50.5	12.2	**34.3**	**31.8**	24.9
GRIP 16B	$\mathbf{40.9 \pm 0.4}$	**65.9**	56.9	42.7	**51.5**	**15.9**	34.2	31.5	**28.5**

Table 5: Absolute scores for the 100B-token GRIP-8B component ablation. Avg. is mean $\pm$ standard deviation over three paired seeds; task columns are seed means.

Variant	Avg.	HE	HE$^+$	MBPP	MBPP$^+$	LCB	Crux-I	Crux-O	MultiPL-E
Random	26.9 ± 0.3	45.6	44.4	30.2	34.1	2.3	22.1	26.0	10.3
UniGeM	34.3 ± 0.4	53.6	50.2	37.2	46.8	7.5	31.4	28.3	19.4
Static Budget	27.6 ± 0.2	47.4	45.7	31.5	34.3	2.2	22.7	25.4	11.6
Static Replay	29.3 ± 0.3	49.0	47.1	33.0	37.5	3.2	24.4	26.5	13.8
Loss Replay	31.5 ± 0.4	51.1	48.8	35.0	40.9	5.7	27.2	27.1	16.2
Diversity	33.6 ± 0.5	53.8	50.2	36.7	44.4	7.1	28.9	28.1	19.7
GRIP	$\mathbf{35.5 \pm 0.3}$	**55.4**	48.5	36.0	**48.9**	**9.7**	**32.6**	27.7	**25.1**

Table 6: Change in seed-mean percentage points when each component is added to the preceding 100B-token GRIP-8B variant. Avg. is the difference between the displayed average scores.

Added component	Avg.	HE	HE$^+$	MBPP	MBPP$^+$	LCB	Crux-I	Crux-O	MultiPL-E
Static Budget over Random	0.7	1.8	1.2	1.3	0.2	-0.1	0.6	-0.6	1.3
Static Replay over Static Budget	1.7	1.6	1.5	1.5	3.2	1.0	1.6	1.1	2.2
Loss Replay over Static Replay	2.2	2.1	1.7	2.0	3.4	2.5	2.8	0.6	2.4
Diversity over Loss Replay	2.1	2.7	1.4	1.6	3.4	1.4	1.7	1.0	3.5
Length Correction over Diversity	1.9	1.6	-1.7	-0.7	4.6	2.6	3.8	-0.4	5.4

Table 7: Three-seed sensitivity to the length exponent. Confidence intervals and p values compare each positive exponent with $\beta = 0$.

β	Avg.	Δ vs. 0	95% CI	p
0.0	33.6 ± 0.5	–	–	–
0.1	34.7 ± 0.4	1.10	$[-1.11, 3.31]$	0.165
0.3	$\mathbf{35.5 \pm 0.3}$	**1.89**	$[\mathbf{1.39}, \mathbf{2.38}]$	**0.0037**
0.5	35.3 ± 0.5	1.70	$[0.82, 2.59]$	0.014

F LENGTH-DECOUPLING DIAGNOSTIC

F.1 LENGTH DISTRIBUTION AND GEOMETRIC COLLAPSE

Figure 7 contains the two length diagnostics from the main paper. Normalized 10-NN distance declines with original document length after 32K chunking and document-level aggregation, placing long documents in denser embedding regions. The candidate corpus also contains fewer long documents. Their low inverse-density score and low corpus frequency jointly reduce selection from the tail.

F.2 RESIDUAL LENGTH DEPENDENCE

Both panels use the same documents and original token length on the horizontal axis. The raw geometric ratio correlates with length ($r = 0.811$); removing its fitted length trend reduces the correlation to $r = 0.014$. The component ablation reports the associated score changes.

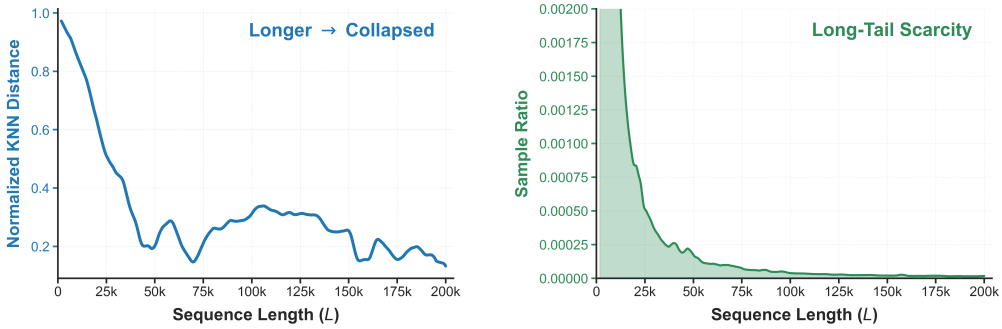


Figure 7: Length diagnostics before rectification. Left: normalized 10-NN distance versus original document length. Right: empirical sample ratio across the same length range. Long documents occupy a geometrically compressed and sparsely populated tail.

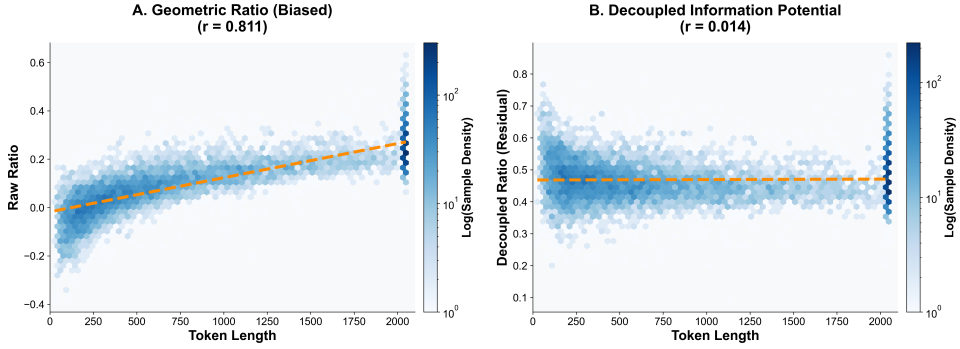


Figure 8: Length dependence of the geometric score after 32K chunking and document-level aggregation. The raw ratio correlates with original document length (left, $r = 0.811$); removing the fitted trend produces a residual with near-zero linear correlation (right, $r = 0.014$).