

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 LAYERMIX LAW: SCALING LAW FOR LARGE LAN- GUAGE MODELS ON QUALITY-WEIGHTED MIXTURE DATA WITH REPETITION

Anonymous authors

Paper under double-blind review

ABSTRACT

Upweighting high-quality data in large language model (LLM) pretraining typically improves performance. However, the limited availability of high-quality data—particularly in overtrained regimes—means that stronger upweighting often increases repetition, which can degrade performance. This creates a fundamental trade-off between data quality and data repetition. In this paper, we systematically investigate how varying data quality and repetition affects models across different scales. Concretely, we partition the source corpus into buckets based on quality scores and sample from each bucket with different weights, thereby constructing training sets with diverse scales, quality distributions, and repetition levels. We then train a family of models on these datasets to measure performance across conditions. Building on these observations, we introduce a theoretical framework analogous to scaling laws, which we call **LayerMix Law**. LayerMix Law predicts model loss as a function of consumed tokens, model size, sampling weights, and repetition levels. The key intuition is to view training as the accumulation of information from data, where the amount of information is governed by data quality, while model scale and repetition determine the information gained per training step. We show that LayerMix Law accurately predicts the model performance on unseen data recipes at larger computation scale (up to 7B parameter run with 425B token, each x2 invest compute), with 0.15% average absolute error and 0.96% maximum absolute error, which enables efficient search for optimal data recipes without costly additional experiments. Moreover, LayerMix Law extrapolates reliably to different degrees of overtraining, providing a efficient tool for selecting data recipes under varying computational budgets.

1 INTRODUCTION

Training large language models (LLMs) requires access to high-quality data (Brown et al., 2020a; Chowdhery et al., 2023). However, the availability of high-quality data is severely limited (Villalobos et al., 2024), and in the data-constrained settings, upweighting higher-quality data inevitably increases repetition, which has been shown to impair performance when excessive (Muennighoff et al., 2023). This issue is further exacerbated by the widespread adoption of overtraining (Touvron et al., 2023; Yang et al., 2025)—a strategy that reduces inference costs compared to the compute-optimal regime (Hoffmann et al., 2022).

To address the shortage of high-quality data as model scale increases, a common compromise is to incorporate lower-quality data, thereby reducing the repetition of high-quality samples. Intuitively, high-quality data provides greater performance gains than low-quality data upon first exposure, but as repetition increases, the marginal benefit decays—eventually approaching that of unseen low-quality data. However, the optimal balance between quality and repetition remains unclear. A standard approach for identifying optimal mixing strategies is to run smaller-scale experiments and extrapolate performance to larger compute budgets using scaling laws (OpenAI et al., 2024; Hoffmann et al., 2022; Chowdhery et al., 2023). Yet, as shown in Figure 1, under conditions of data repetition, standard scaling laws fail to reliably predict model performance at scale (Hernandez et al., 2022; Muennighoff et al., 2023). Moreover, they do not generalize across different mixing

054 strategies , necessitating grid searches over data recipes—an approach that is costly even at small
055 scales.

056 In this paper, we study the problem of scaling large language models in a data-aware regime, where
057 training data consists of a heterogeneous mixture with varying quality levels, and each quality level
058 is repeated to different extents. We introduce a theoretical framework, the LayerMix Law, which
059 accounts for both the scaling effects of mixture weights and the impact of repetition. Our formulation
060 views training as a process of accumulating information from the dataset, with model performance
061 determined by the total information gained by the end of training. At each step, the information gain
062 is modeled as the sum of contributions from different quality ranges. Within each quality range,
063 the gain depends on two factors: an information density function, parameterized by quality (with
064 higher quality assigned higher density), and an exponential decay term that captures the interactions
065 between model scale, data scale, and repetition level.

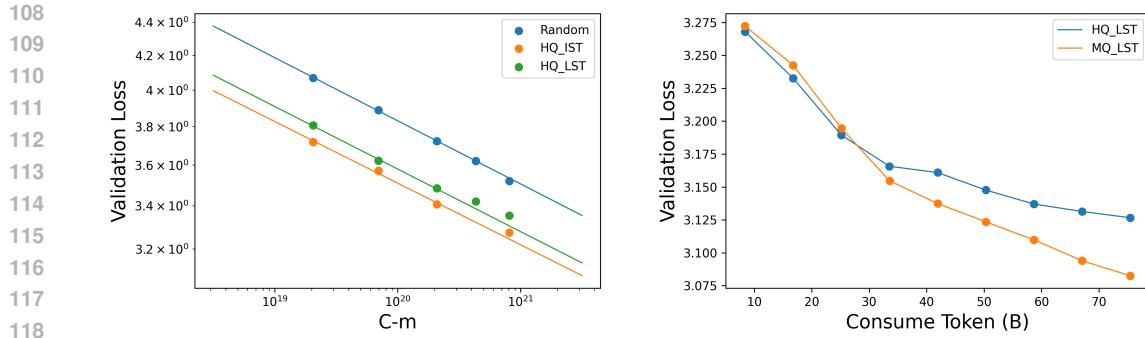
066 To fit the parameters of the LayerMix Law, we construct a suite of datasets that vary along three
067 axes: scale, quality, and repetition level. Specifically, we partition the source dataset into buckets
068 according to quality scores, and then sample from each bucket with different weights, a procedure
069 we refer to as LayerMix sampling. Following the data-constrained setting, the source dataset is first
070 downsampled to the target scale to ensure stable repetition effects. We then train 9 models ranging
071 from 252M to 1.2B parameters from scratch, each under the same 3.6x over-trained ratio (Gadre
072 et al., 2024). For each model, we construct three datasets with distinct LayerMix sampling configu-
073 rations, resulting in 27 total training runs. Model performance is evaluated as the average perplexity
074 across five downstream tasks. Finally, we fit the LayerMix Law to these results, estimating the
075 parameters that best capture the relationship between information gain and observed performance.

076 To verify the generalization ability of LayerMix Law, we conduct three extrapolate experiments.
077 Firstly we create datasets with two other unseen LayerMix sampling parameters to test the extrap-
078 olation on unseen data recipe. Secondly, we conduct experiments with larger computation scale on
079 both seen and unseen LayerMix sampling parameters to test the scaling ability. Thirdly, we conduct
080 experiments with larger over-trained ratio (25x), to test the generalization ability of LayerMix Law
081 parameters. We found that LayerMix Law is able to predict loss on unseen data recipes at differ-
082 ent scales (up to 7B parameter run with 425B token, each x2 invest compute), with 0.15% average
083 absolute error and 0.96% maximum absolute error. We search for the optimal data recipe on 2.5B
084 model with LayerMix Law with no additional experiments, and the result achieves the best among
085 4 other random ablations. Also on over-trained ratio, directly apply LayerMix Law yields good
086 scaling results.

088 2 RELATED WORK

089 **Scaling Laws** Scaling of transformer language models (Vaswani et al., 2017) and training data
090 has been shown to provide consistent performance improvements (Chowdhery et al., 2023; Radford
091 et al., 2019). This has led to the development of many large-scale models, including both dense
092 architectures (Brown et al., 2020b; Rae et al., 2021; Grattafiori et al., 2024) and mixture-of-experts
093 (MoE) variants (DeepSeek-AI et al., 2025; Yang et al., 2025; Fedus et al., 2021). Early empiri-
094 cal studies observed that neural networks exhibit predictable power-law scaling behavior (Hestness
095 et al., 2017). Building on this, Hoffmann et al. (2022) investigated the compute-optimal setting, sug-
096 gesting that model size and training data should be scaled in roughly equal proportions, whereas Ka-
097 plan et al. (2020) proposed a different allocation strategy emphasizing alternative trade-offs. More
098 recent work, such as DeepSeek-AI et al. (2024), further explored how compute budget C interacts
099 with optimization hyperparameters, including the choice of batch size and learning rate.

100 Recently, there has been a growing trend of over-training smaller models on large datasets (Tou-
101 vron et al., 2023; Yang et al., 2025), motivated by both efficiency considerations and deployment
102 constraints. Sardana et al. (2024) extended the Chinchilla framework by incorporating data quality
103 and inference requirements, deriving optimal allocations between model size and dataset scale under
104 these additional factors. Complementary to this, Gadre et al. (2024) demonstrated that scaling laws
105 remain reliable even in the over-trained regime, where models are trained significantly beyond the
106 compute-optimal point. These findings highlight the importance of revisiting scaling strategies in
107 regimes constrained by data availability, quality variation, or inference efficiency demands.



(a) *Loss- C_m* Curve of model trained with random data and Layermix packed data

(b) The validation loss of different LayerMix experiments in training process.

Figure 1: Effects of quality selection and data repetition

For predicting downstream performance, Isik et al. (2025) demonstrated that downstream metrics also exhibit predictable scaling effects after fine-tuning, extending the scope of scaling laws beyond pre-training loss. Schaeffer et al. (2023) further established a connection between non-linear evaluation metrics and model perplexity, providing a more stable predictor of performance compared to prior approaches such as Wei et al. (2022), which observed instability in emergent metrics. Ge et al. (2025) employed fine-grained alignment between a model’s foundational capabilities and the requirements of specific downstream tasks, leading to more accurate scaling law predictions for that task. Ruan et al. (2024) leveraged performance data from existing models to predict complex behaviors and emergent abilities by modeling them in a shared, low-dimensional capability space. Both approaches highlight a growing trend toward understanding scaling phenomena through the lens of modular capabilities.

Data-Aware Scaling Traditional scaling laws typically assume that training data is both fixed and unlimited. In practice, however, high-quality data is scarce and often upsampled to improve model performance (Lin et al., 2022). Xue et al. (2023) found that continuing to train on repeated data is generally preferable to stopping early. This aligns with earlier findings by Hernandez et al. (2022) and Muennighoff et al. (2023), who also observed performance deterioration when upsampling or repeating datasets. These insights underscore the limitations of classical scaling laws in data-constrained settings and highlight the need for more sophisticated, data-aware approaches. More recently, Chen et al. (2025) studied how sub-scaling laws interact with data density, providing a finer-grained understanding of scaling under limited-data regimes.

Other lines of work have applied scaling laws to guide the design of optimal data recipes. For instance, Ye et al. (2025) incorporated data mixture weights into the functional form of loss prediction, while Kang et al. (2025) suggested that optimal mixing strategies themselves depend on model scale. Liu et al. (2024) employed a proxy model to predict the final model’s performance under different mixture ratios, thereby discovering high-quality data recipes without training large-scale models. Gu et al. (2024); Que et al. (2024) investigated scaling laws in the context of continued pre-training, using them to inform domain mixture strategies. In addition, Chang et al. (2024) analyzed how scaling interacts with data quality. In contrast to these approaches, our work aims to predict model loss on mixtures of varying quality and repetition levels, thereby providing a more general framework for data-aware scaling.

3 LIMITATIONS OF CONVENTIONAL SCALING LAWS

In this section, we reveal and substantiate a critical limitation of conventional scaling laws in the context of data repetition and quality selection. First, we introduce the LayerMix sampling function in section 3.1, to imitate real scenario where the data is a mixture of different quality and repetition degrees. Next, we compare the relationship between the model’s loss L and amount of compute C in cases with and without repetition in section 3.2, and the results show that the traditional scaling law performs well on data without repetition

162 Table 1: Preset LayerMix sampling weights and Searched optimal sampling weights for 2.5B model.
163

Name	w1	w2	w3	w4	w5	w6
HQ (High Quality)	0.80	0.10	0.03	0.03	0.02	0.0
MHQ (Medium-High Quality)	0.66	0.22	0.05	0.03	0.02	0.0
MQ (Medium Quality)	0.48	0.23	0.13	0.07	0.07	0.0
MLQ (Medium-Low Quality)	0.38	0.21	0.20	0.11	0.08	0.0
LQ (Low Quality)	0.24	0.20	0.19	0.18	0.17	0.0
Optimal Recipe of 2.5B model with $m = 3.6$	0.50	0.49	0.01	0.0	0.0	0.0

171
172
173 3.1 LAYERMIX SAMPLING FUNCTION
174175 **Source Data** We obtain our training corpora from Common Crawl (Common Crawl Foundation),
176 following similar filtering process as Penedo et al. (2023) and obtain 15T English tokens. We ran
177 global fuzzy deduplication across all snapshots to ensure there is no repeat data in the corpora. The
178 final dataset contains 3.7T token. The details are in Appendix A.
179180 **Training Data Sampling** We first define the quality score for each document following Liu et al.
181 (2025), where two quality classifiers (Penedo et al., 2024; Li et al., 2025) are applied and the final
182 quality score is averaging the normalized score from each classifier. With the quality score, We then
183 rank all documents based on quality score and empirically split the training corpora into six quality
184 buckets based on their quality rankings, which are 0-5%, 5%-20%, 20%-40%, 40%-60%, 60%-80%,
185 80%-100%.186 Then we define a LayerMix sampling function $H(w, K, S, B)$, where S represents the packing
187 source tokens and K represents the packed tokens for training data, and we perform 1 epoch
188 training to avoid additional repetition factors. $w = [w_0, w_1, \dots, w_5]$ and $\sum w_d = 1$, repre-
189 senting the token proportion of the data in the d -th bucket in the packed training data, and
190 $B = [B_0, B_1, \dots, B_5]$ is the token proportion of source token for each data bucket, in our exper-
191 iments, $B = [5\%, 15\%, 20\%, 20\%, 20\%, 20\%]$. The LayerMix sampling function H returns a dataset
192 of total token K with bucket proportions given by w sampled from a corpora of total token S . In
193 the returned training data, the d -th bucket have $w_d K$ tokens, which are sampled from $B_d S$ tokens
194 in source data, leading to $M_d = \min(w_d K, B_d S)$ unrepeat tokens. Then the average repeat time
195 for data in the d -th bucket is $R_d = \frac{M_d}{B_d S}$. The detailed algorithm is shown in Appendix B.196 With different setting of w, K, S , the LayerMix sampling function generate datasets with different
197 scales, quality and degrees of repetition. We ensure w_d is greater than w_{d+1} to guarantee higher
198 quality buckets have a larger proportion in the training data. We select 5 settings of w , representing
199 different levels of quality and repetition, namely HQ (High Quality), MHQ(Medium-High Quality),
200 MQ (Medium Quality), MLQ (Medium-Low Quality), LQ (Low Quality). The detailed proportion
201 is as Table 1. w_5 is set to 0 so that we drop the bottom 20% of data ranked by quality score.
202 Throughout work, we set $K = S$ to reduce complexity unless mentioned, so that we only focus on
203 the repetition change caused by w .
204205 3.2 TRADITIONAL SCALING LAW BETWEEN LOSS AND AMOUNT OF COMPUTE
206207 We compare the relationship between model loss L and total compute C under regimes with and
208 without repetition in an overtrained setting. Specifically, under the compute-optimal scheme, $C_{opt} =$
209 $N_{opt} K_{opt}$, where K is the consume token, N is the non-embedding FLOPs per token as defined in
210 DeepSeek-AI et al. (2024) and N_{opt} , K_{opt} is the Chinchilla-optimal pair. Then in the overtrained
211 setting, following Gadre et al. (2024), we set $K_m = \sqrt{m} K_{opt}$, $N_m = \frac{1}{\sqrt{m}} N_{opt}$, $C_m = K_m N_m$
212 with $m = 3.6$. And Gadre et al. (2024) shows that the the Loss–Compute relation preserves the
213 fitted exponent for models trained with the same overtraining factor m .
214215 **Random** The training data is randomly sampled from sufficient large source data, where $S \gg K$,
meaning there is rarely no repetition in the training data.

216 **HQ_IST** We use the layermix sampling function and w is set to mainly focus on high quality data,
 217 namely HQ (High Quality), see details in Appendix B. Then we set $S \gg K$, denoting as IST
 218 (Infinite Source Token), meaning there is almost no repetition in train data

219 **HQ_LST** We use the layermix sampling function and set $S = K$, denoting as LST (Limited Source
 220 Token), where there exists repetition in the training data.

222 We plot the log-log plot between C_m and model loss L in Figure 1a. Note that the loss L mentioned
 223 here, including later references, is the average loss of the model on the following five downstream
 224 tasks: HellaSwag (Zellers et al., 2019), ARC-E, ARC-C (Clark et al., 2018), MMLU (Hendrycks
 225 et al., 2021), TriviaQA (Joshi et al., 2017). Following Schaeffer et al. (2023), we transfer the accu-
 226 racy on downstream tasks into perplexity for better scaling effect. The results show that for random
 227 dataset, the scaling law loss-C curve can fit all of the data points. But for **HQ_IST** and **HQ_LST**, the
 228 performance decay as compute C_m scales. More results are detailed in Appendix F

229 The above observation indicating that traditional scaling law only holds for completely random and
 230 non-repeated data. The change of data quality distribution or repeat time undermines its predictive
 231 accuracy. So we need a modified scaling law that incorporates both data quality distribution and the
 232 degree of data repetition as core variables.

234 4 LAYERMIX LAW

236 Scaling laws for large language models traditionally rely on compute, but often fail to account
 237 for effects of data quality and repetition. In this section, we introduce the design of LayerMix
 238 Law. We treat the training process as gaining information from the dataset and propose to calculate
 239 Information Quantity as accumulation of information gain throughout the training process, which
 240 synthesizes the impacts of data quality, repetition level, model scales and total training tokens, and
 241 then build power-law relationship with the model’s final validation loss.

243 4.1 INFORMATION QUANTITY

245 We first show how the evaluation loss changes during training of two 850M models trained on two
 246 datasets packed with different LayerMix sampling weights in Figure 1b. The dataset **HQ_LST** re-
 247 peates top 5% quality data for about 16 times and **MQ_LST** for 10 times, **MQ_LST has lower quality**
 248 **than HQ_LST but less repetition**. By default, we use the LST setting and we ignore the notation of
 249 LST for simplicity unless mentioned. In the early training stage, **HQ** and **MQ** experiments achieves
 250 almost same evaluation loss. However, later in the training, the loss of **HQ** experiment decreases
 251 much more slowly, resulting in a worse final performance compared to the **MQ** experiment, indicat-
 252 ing that more repetition expedites the decay of performance gaining.

253 Based on this observation, we propose an exponential decay function to model the decreasing infor-
 254 mation gain of repeated data. Assuming the Information Quantity a document i contains is I_i , then
 255 the information a language model gets at t -th learning from the document i is:

$$257 \quad I_{i,\text{part}}(t, \lambda_N) = I_i \cdot \lambda_N e^{-\lambda_N t} \quad (1)$$

259 where λ is a hyperparameter which is related to the non-embedding FLOPs/token N .

261 When a language model learning the document for total T times, the Information Quantity learned
 262 from the document is:

$$264 \quad I_{i,\text{total}}(T, \lambda_N) = \int_0^T I_{i,\text{part}}(t, \lambda_N) dt = I_i \cdot (1 - e^{-\lambda_N T}) \quad (2)$$

268 Equation 2 captures the principle of diminishing returns in learning: repeated exposure to a doc-
 269 ument yields progressively smaller gains, causing the total acquired information to saturate and
 asymptotically approach the document’s full information content I_i .

Considering in large-scale training, due to the forgetting effect: as the total train token K grows, the average Information Quantity language model acquire from a singel sample decreases. We introduce train token K to the Equation 1 to accommodate this phenomenon:

$$I_{i,\text{part}}(t, \lambda_N, K) = I_i \cdot \lambda_N e^{-\lambda_N t / \log(K)} \quad (3)$$

Then the Equation 2 becomes to:

$$I_{i,\text{total}}(t, \lambda_N, K) = \int_0^T I_{i,\text{part}}(t, \lambda_N, K) dt = I_i \cdot \log(K) (1 - e^{-\lambda_N T / \log(K)}) \quad (4)$$

For all the training data, we sum them together as the final Information Quantity the language model learned from the training corpora, denoting as $info$:

$$\begin{aligned} info(w, K, S, f, \lambda_N) &= \sum_d I_d \cdot \log(K) (1 - e^{-\lambda_N R_d / \log(K)}) \\ &= \sum_d f_d M_d \log(K) \cdot (1 - e^{-\lambda_N R_d / \log(K)}) \end{aligned} \quad (5)$$

where d is the quality bucket number from 0 to 5. I_d is the total information quantity in d -the bucket, which can be calculated by the multiplication of number of unique tokens $M_d = \min(w_d K, B_d S)$ and information density f_d , which is a parameterized quality density function. $R_d = \frac{w_d K}{M_d}$ is the average repeat times for the data from the d -th bucket and λ_N is related with N , which are to be fitted from the data.

Equation 5 can be divided into two parts: the first term is $I_d = f_d M_d \log(K)$, it represents the total Information Quantity contained in the packed data of the d -th bucket, and the second term is $1 - e^{-\lambda_N R_d / \log(K)}$, it represents the language model's learning ability on this data when repeated an average of R_d times. And the total Information Quantity learned by the language model is the product of these two terms.

We propose Information Quantity, a metric computed from LayerMix sampling weights w , train token K and two fitted functions (f_d, λ_N) , to quantify the knowledge learned during training. Since it is designed to be monotonic with model performance, it enables loss prediction for various training configurations prior to any actual runs. The fitting of f_d and λ_N is described in Section 5.2.

4.2 INFORMATION-LOSS SCALING LAW ON REPEATED DATA

As illustrated in Figure 1a, the evaluation loss of the model trained with repeated data is higher than that of the model trained without repeated data, and the traditional scaling law fails to predict language model's performance under this circumstance.

We use the Information Quantity proposed in Section 4.1 and plot the L - $info$ figure. As illustrated in Figure 2, when we replace the traditional computation axis C with our novel metric: Information Quantity, the experimental points with different LayerMix sampling weights w , Model non-embedding FLOPs/token N and Train Token K now collapse perfectly onto a single, unified power-law curve, where they were previously scattered and separated.

Then the relationship between the loss L and $info$ can be measured using power-law formulation as:

$$L = \alpha \cdot info^{-\beta} \quad (6)$$

In our experiment, $\alpha = 3.7373$ and $\beta = 0.0441$. We show them in a log-log plot, so it appears as a straight line with a slope of $-\beta$ and an intercept of $\log(\alpha)$.

Like the traditional scaling law (Hoffmann et al., 2022), we can now conduct experiments on small models to compare the advantages and disadvantages of different experimental configurations, and

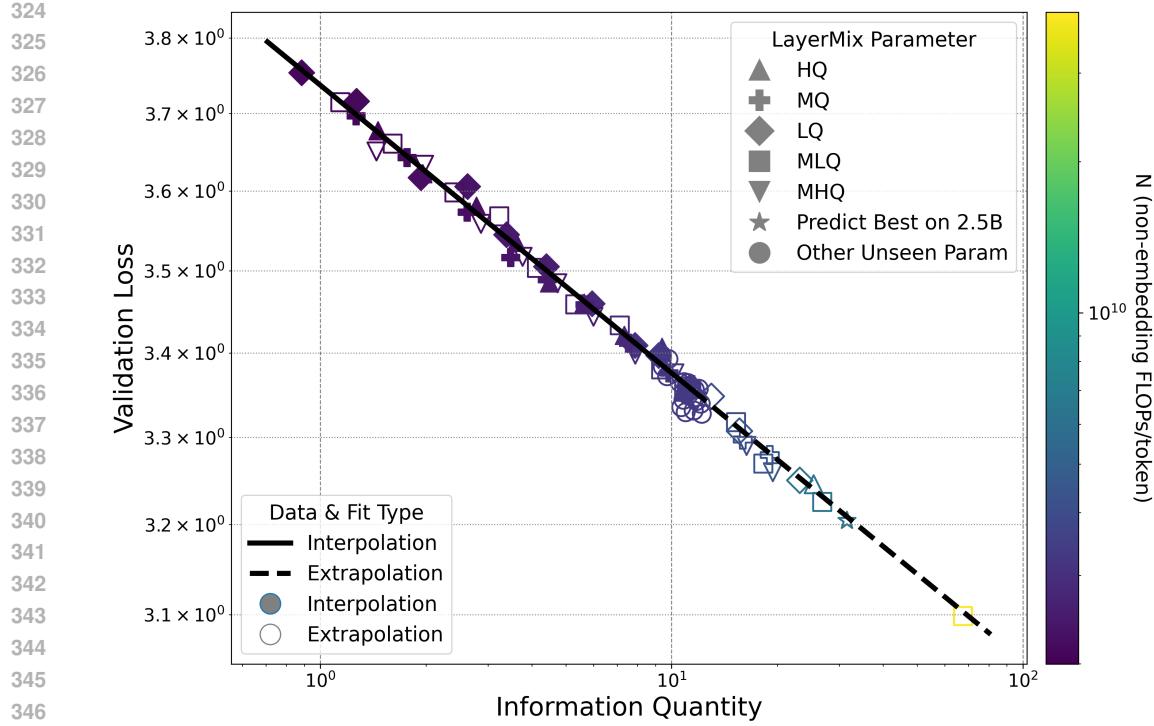


Figure 2: The Unified Information-Loss Scaling Law. The scaling law is fitted only on the interpolation data (solid markers), yet it accurately predicts the performance of held-out extrapolation data (hollow markers).

then use our proposed information scaling law to extrapolate the performance of larger models under larger training tokens.

5 FITTING EXPERIMENTS

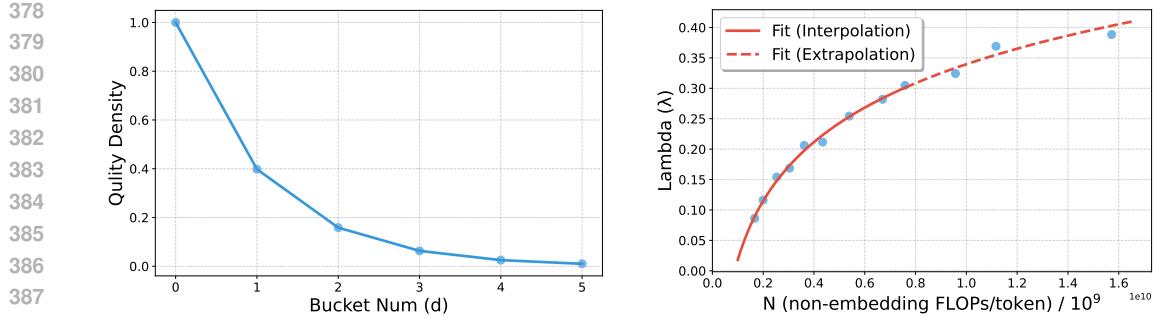
5.1 TRAINING SETUP

We train 9 models ranging from 252M to 1.2B on 3 layermix sampling weights (*HQ*, *MQ*, and *LQ*), with 3.6x over-trained ratio, resulting in 27 experiment runs in total to collect data for fitting the LayerMix Law parameters. We use transformer architecture (Vaswani et al., 2017), SwiGLU (Shazeer, 2020) as the activation function and RoPE embeddings (Su et al., 2024). We use a tokenizer with 250k vocabulary. See Appendix B and Appendix C for details about LayerMix sampling weights, model structure, learnign rate and optimizer.

5.2 FITTING THE CURVE

In this section, we introduce how to fit the parameters in LayerMix Law to predict the model performance collected in Section 5.1. Since Information Quantity $info$ indicates the knowledge learned by the model, we expect larger $info$ to correspond to lower evaluation loss L . Considering that there may exist scale difference between $info$ and model loss L , we choose Spearman correlation ρ_s as the fitting metric, i.e., the object is to find the optimal quality density f and λ_N such that the Spearman correlation between evaluation loss L and $info$ is minimized for all the experiments over N, w :

$$(f^*, \lambda^*) = \operatorname{argmin}_{f, \lambda} \sum_{f, \lambda}^N \sum_w^w (\rho_s (L_N, info(w, K_N, S_N, f, \lambda_N))) \quad (7)$$



(a) The fitted quality density function f_d . The quality density is a monotonically decreasing function of the bucket index, meaning buckets with higher-quality data are assigned a higher density value.

(b) The relationship between λ and N with a fitted curve. The blue scattered points represent the observed data. The solid red line shows the fit within the data range, while the dashed line represents the extrapolation.

Figure 3: The fitted function of quality density function and relationship between λ_N and N

To prevent from over-fitting, we make some assumption based on naive intuition. For f , as it indicates the quality density, the higher-quality bucket should have larger f . As smaller d corresponds to higher-quality buckets, we define f in the following form to ensure it is a decreasing function:

$$f_d(\theta) = e^{-\theta*d} \quad (8)$$

where θ is a hyperparameter and $\theta > 0$.

λ_N is related to the model's learning capacity, so λ_N should increase as N increases. But we need to find the formula for λ_N related with N so that it can scale to larger N . To do this we first sample 100,000 combinations of θ and λ_N from the parameter space, then select optimal θ^* and λ_N^* based on Equation 7. The fitted quality density $f(\theta^*)$ is shown in Figure 3a with fitted $\theta^* = -0.922$.

Having the λ_N^* values of different models, as is shown in Figure 3b, we try to fit the λ_N - N curve. The relationship between λ_N and N is observed to be non-linear, exhibiting rapid growth for smaller N and gradually saturating as N increases. This trend is well-approximated by a logarithmic function. Therefore, we choose the λ_N - N curve using following formula:

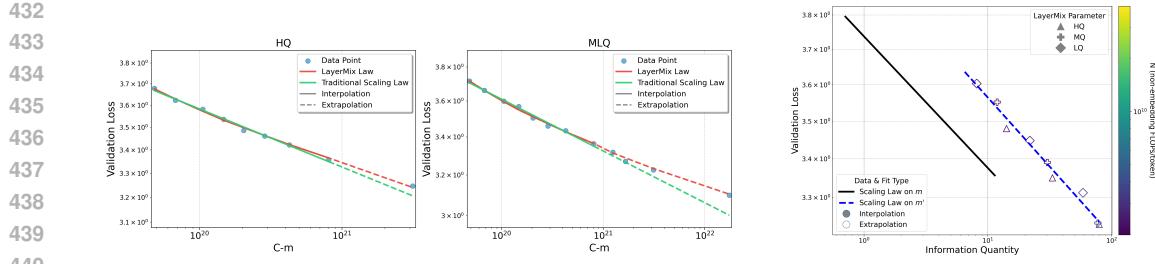
$$\lambda_N(a, b) = a \cdot \ln(N) + b \quad (9)$$

Using existing λ_N^* , we fit the λ_N - N curve in Figure 3b with fitted $a^* = 0.140$, $b^* = 0.018$. To validate this fit, we compute λ_N^* for larger N under the fixed θ^* , and examine whether these values lie on the predicted λ_N - N curve. As illustrated in Figure 3b, the results demonstrate strong extrapolation performance, supporting the correctness of our formulation. **We compared with different formats of 9 in Appendix E and the log function best fit the trend and extrapolates**

Finally, with $f(\theta^*)$ and $\lambda_N(a^*, b^*)$, we can calculate the Information Quantity for arbitrary layermix sampling weights w , train token K , source token S and model non-embedding FLOPs/token N .

6 EXTRAPOLATION

The LayerMix Law achieves a strong fit on the training data after parameter optimization, and we subsequently employ it to predict loss under the unseen conditions to assess its robustness. To rigorously evaluate its capability, we first compare it with the traditional scaling laws. Then we test the extrapolation along three key axes: (i) novel LayerMix sampling weights, (ii) larger computational scales, and (iii) varying degrees of over-training. Finally, we use our LayerMix Law to predict optimal data recipe under different training settings and validate the recipe by comparing with preset recipes.



(a) Validation loss versus compute C in the loss– C view. Curves are fit on 252M–1.2B and extrapolated to larger models.

(b) Cross-Regime Prediction of the Scaling Law.

Figure 4: The extrapolation results of LayerMix Law

Comparing with traditional scaling laws

Figure 4a contrasts our LayerMix law with the traditional power scaling law in the loss– C plane. Both curves are fit using models in the 252M–1.2B range and then extrapolated to larger models. The Info curve tracks the MLQ data more closely within the fitting regime and remains accurate when extrapolating up to 7B models, avoiding the overly optimistic loss reductions predicted by the traditional law at high compute. Concretely, the traditional scaling law tends to under-estimate loss as C_m grows, whereas the Info curve preserves a data-aligned decay that better matches the realized validation losses of larger models.

Extrapolation to other LayerMix Sampling Weights

We first test the ability to generalize to an unseen LayerMix sampling weights. We test on unseen dataset generated with MLQ , MHQ on model scales ranging from 252M to 1.2B, which are within the range of training data. Also we random sample 25 more sampling weights and run experiments on 1.2B model only.

The result is shown in Figure 2 marked by hollow squares (for MLQ) and hollow downward-pointing triangles (for MHQ). As can be seen, these points align remarkably well with the scaling law curve established by the initial HQ , MQ , LQ data, demonstrating the predictive power of our model on unseen LayerMix sampling weights. The traditional scaling laws requires additional experiments on different data recipes to fit new curves, while ours can directly predict loss on unseen recipes.

Extrapolation to Larger Models

To test the extrapolation ability on model scale, we use the same Layermix sampling weights MQ, LQ to train models ranging from 1.5B to 2.5B and HQ, LQ to train model with 2.5B parameters, which are out of the range of training data. The experimental results of larger models are shown in Figure 2, we can see LayerMix Law predict the loss on larger scale accurately for all three sampling weights, proving the ability of scaling on model size.

Combination of Extrapolation

Further more, we combine the two extrapolation above and test the effectiveness on both unseen LayerMix sampling weights and unseen scales. We run experiments with MLQ, MHQ on models ranging from 1.5B to 7B. As shown in Figure 2, LayerMix Law also generalise well on these combined extrapolation condition. On all the unseen data points, including unseen LayerMix sampling weights and model scales, LayerMix Law predict the validation loss with 0.15% average absolute error and maximum error is 0.96%. This proves that our proposed information scaling law has reliable extrapolation capability.

Extrapolation to Larger Overtrain Degree

To explore the model’s reliability under varying sub-optimality, we conducted a second series of experiments at a higher overtrain degree, $m' = 25$. This new regime was anchored by a 1.2B model trained on 640B tokens (the $C_{m'}$ experiment), contrasting with our initial C_m experiment anchored at 106B tokens.

486
487
488 Table 2: Results of four distinct LayerMix experiments on 2.5B model.
489
490
491

Name	HQ	LQ	MLQ	Pred Best
Validation Loss	3.246	3.250	3.226	3.204

492
493 Table 3: The best data recipe for different models and train token
494

Model	Train Token	Source Token	w1	w2	w3	w4	w5	w6
7B	500B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	800B	500B	0.439	0.430	0.130	0.001	0.000	0.000
	1000B	500B	0.395	0.387	0.214	0.003	0.001	0.000
1.8B	500B	500B	0.548	0.444	0.004	0.003	0.002	0.000
	800B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	1000B	500B	0.491	0.487	0.017	0.005	0.000	0.000
1.2B	500B	500B	0.619	0.376	0.004	0.001	0.000	0.000
	800B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	1000B	500B	0.496	0.492	0.007	0.003	0.002	0.000

503
504
505
506 For the $C_{m'}$ -experiment, we calculated the Information Quantity using the same quality density
507 $f(\theta^*)$ and $\lambda_N(a^*, b^*)$ fitted previously on the C_m data. As shown in Figure 4b, the new experimental
508 points align with a new scaling law curve. The resulting curves for C_m and $C_{m'}$ appear nearly
509 parallel, suggesting the overtrain degree m primarily shifts the curve’s intercept. This confirms that
510 our proposed Information Scaling Law is effective across different overtrain degrees.

511 Optimizing Data Recipe with LayerMix Law

512 The ability of predicting loss on unseen data recipes and scales enables us to search for best data
513 recipe without additional experiments. Similar to Liu et al. (2024). We randomly sample 100k
514 LayerMix parameters from the parameter space, compute the information for each set of parameters,
515 and convert it to loss via Equation 6. We then select the parameter that minimizes the predicted
516 validation loss as the optimal LayerMix configuration for each training setting.

517 To verify the optimal recipe, we conduct experiments on 2.5B model with optimal data recipe and
518 3 other layermix sampling weights. The result optimal recipe is as in Table 1. As shown in Table 2,
519 our optimal recipe achieves the best validation loss.

520
521 In Table 3, we present the optimal LayerMix parameters for different model sizes and training-token
522 counts under a fixed source-token budget of 500B tokens. The optimal LayerMix parameters exhibit
523 two clear trends. First, at a fixed training-token count, smaller models favor a higher fraction of
524 high-quality data, whereas larger models benefit more from diversity and thus allocate a smaller
525 fraction to the high-quality data. Second, as the total training tokens increase, the optimal LayerMix
526 parameters shift from a high-quality emphasis toward greater diversity. More results are shown in
527 Appendix H. In short: Small models or small training budgets prioritize quality; large models or
528 large training budgets prioritize diversity.

530 7 CONCLUSION

531
532
533 In this paper, we propose a refined scaling law modeling **LayerMix Law**, which focus on predicting
534 model performance on downstream tasks under data-constrained settings with weighted-quality
535 mixing. The LayerMix Law provides accurate predictions of model performance on unseen data
536 recipes at larger computational scales, achieving an average absolute error of only 0.15% and a
537 maximum error of 0.96%. This enables efficient discovery of optimal data recipes without the need
538 for extensive additional experiments. Furthermore, the LayerMix Law extrapolates reliably across
539 varying degrees of over-training, offering an effective tool for selecting data recipes under different
computational budgets.

540 8 ETHICS STATEMENT
541542 Our research is based on the publicly available Common Crawl dataset. We do not foresee any direct
543 negative societal impacts stemming from our methodology or the resulting models.
544545 9 REPRODUCIBILITY STATEMENT
546548 Our experiments are based on the open-source Common Crawl dataset. All experimental settings,
549 model architectures, hyperparameters, and implementation details have been thoroughly described
550 in the main body and the appendix to ensure that other researchers can independently reproduce our
551 results based on this information.
552

553 REFERENCES

554 Xiao Bi, Deli Chen, Guanting Chen, Shanhua Chen, Damai Dai, Chengqi Deng, Honghui Ding,
555 Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with
556 longtermism. *arXiv preprint arXiv:2401.02954*, 2024.557 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and et al. Language mod-
558 els are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and
559 H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901,
560 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.561 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,
562 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. Language mod-
563 els are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin
564 (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran
565 Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.566 Ernie Chang, Matteo Paltenghi, Yang Li, Pin-Jie Lin, Changsheng Zhao, Patrick Huber, Zechun Liu,
567 Rastislav Rabatin, Yangyang Shi, and Vikas Chandra. Scaling parameter-constrained language
568 models with quality data, 2024. URL <https://arxiv.org/abs/2410.03083>.569 Zhengyu Chen, Siqi Wang, Teng Xiao, Yudong Wang, Shiqi Chen, Xunliang Cai, Junxian He, and
570 Jingang Wang. Sub-scaling laws: On the role of data density and training strategies in llms, 2025.
571 URL <https://arxiv.org/abs/2507.10613>.572 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, and et al.
573 Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24
574 (240):1–113, 2023. URL <http://jmlr.org/papers/v24/22-1144.html>.575 Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and
576 Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge,
577 2018. URL <https://arxiv.org/abs/1803.05457>.578 Common Crawl Foundation. Common Crawl. <http://commoncrawl.org>.584 DeepSeek-AI, ;, Xiao Bi, Deli Chen, Guanting Chen, Shanhua Chen, Damai Dai, Chengqi Deng,
585 Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, and et al.
586 Deepseek llm: Scaling open-source language models with longtermism, 2024. URL <https://arxiv.org/abs/2401.02954>.588 DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Cheng-
589 gang Zhao, Chengqi Deng, Chenyu Zhang, and et al. Deepseek-v3 technical report, 2025. URL
590 <https://arxiv.org/abs/2412.19437>.592 William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion param-
593 eter models with simple and efficient sparsity. *CoRR*, abs/2101.03961, 2021. URL <https://arxiv.org/abs/2101.03961>.

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

597

598 Qiming Ge, Shuhao Xing, Songyang Gao, Yunhua Zhou, Yicheng Zou, Songyang Zhang, Zhi Chen,
 599 Hang Yan, Qi Zhang, Qipeng Guo, et al. Capability salience vector: Fine-grained alignment of
 600 loss and capabilities for downstream task scaling law. *arXiv preprint arXiv:2506.13216*, 2025.

601 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
 602 Al-Dahle, Aiesha Letman, Akhil Mathur, and et al. The llama 3 herd of models, 2024. URL
 603 <https://arxiv.org/abs/2407.21783>.

604

605 Jiawei Gu, Zacc Yang, Chuanghao Ding, Rui Zhao, and Fei Tan. Cmr scaling law: Predicting
 606 critical mixture ratios for continual pre-training of language models, 2024. URL <https://arxiv.org/abs/2407.17467>.

607

608 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and
 609 Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*. Open-
 610 Review.net, 2021. URL <http://dblp.uni-trier.de/db/conf/iclr/iclr2021.html#HendrycksBBZMSS21>.

611

612 Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk,
 613 Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann,
 614 Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCand-
 615 dish. Scaling laws and interpretability of learning from repeated data, 2022. URL <https://arxiv.org/abs/2205.10487>.

616

617 Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianine-
 618 jad, Md. Mostafa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable,
 619 empirically. *CoRR*, abs/1712.00409, 2017. URL <http://arxiv.org/abs/1712.00409>.

620

621 Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza
 622 Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hen-
 623 nighan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy,
 624 Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre.
 625 Training compute-optimal large language models. In *Proceedings of the 36th International Con-
 626 ference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran
 627 Associates Inc. ISBN 9781713871088.

628

629 Berivan Isik, Natalia Ponomareva, Hussein Hazimeh, Dimitris Paparas, Sergei Vassilvitskii, and
 630 Sammi Koyejo. Scaling laws for downstream task performance in machine translation, 2025.
 631 URL <https://arxiv.org/abs/2402.04177>.

632

633 Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly
 634 supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan
 635 (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics
 636 (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Com-
 637 putational Linguistics. doi: 10.18653/v1/P17-1147. URL <https://aclanthology.org/P17-1147/>.

638

639 Feiyang Kang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia.
 640 Autoscale: Scale-aware data mixing for pre-training llms, 2025. URL <https://arxiv.org/abs/2407.20177>.

641

642 Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child,
 643 Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language
 644 models. *CoRR*, abs/2001.08361, 2020. URL <https://arxiv.org/abs/2001.08361>.

645

646 Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash
 647 Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, and et al. Datacomp-lm: In search of the next
 648 generation of training sets for language models, 2025. URL <https://arxiv.org/abs/2406.11794>.

648 Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, and
 649 et al. Few-shot learning with multilingual generative language models. In Yoav Goldberg, Zor-
 650 nitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods*
 651 *in Natural Language Processing*, pp. 9019–9052, Abu Dhabi, United Arab Emirates, December
 652 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.616. URL
 653 <https://aclanthology.org/2022.emnlp-main.616/>.

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

657

658 Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing
 659 Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv*
 660 *preprint arXiv:2407.01492*, 2024.

661 Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Noua-
 662 mane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language
 663 models. In *Proceedings of the 37th International Conference on Neural Information Processing*
 664 *Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

665

666 OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren-
 667 cia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red
 668 Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, and et al. Gpt-4 technical
 669 report, 2024. URL <https://arxiv.org/abs/2303.08774>.

670 Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli,
 671 Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb
 672 dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023.
 673 URL <https://arxiv.org/abs/2306.01116>.

674

675 Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin
 676 Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the
 677 finest text data at scale, 2024. URL <https://arxiv.org/abs/2406.17557>.

678

679 Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan,
 680 Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, Xu Tan, Jie Fu, Wenbo Su, Jiamang Wang, Lin Qu,
 681 and Bo Zheng. D-cpt law: Domain-specific continual pre-training scaling law for large language
 682 models, 2024. URL <https://arxiv.org/abs/2406.01375>.

683

684 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever.
 685 Language models are unsupervised multitask learners. *OpenAI*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Accessed: 2024-11-15.

686

687 Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis
 688 Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, and
 689 et al. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*,
 690 [abs/2112.11446](https://arxiv.org/abs/2112.11446), 2021. URL <https://arxiv.org/abs/2112.11446>.

691

692 Yangjun Ruan, Chris J Maddison, and Tatsunori B Hashimoto. Observational scaling laws and
 693 the predictability of langauge model performance. *Advances in Neural Information Processing*
 694 *Systems*, 37:15841–15892, 2024.

695

696 Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. Beyond chinchilla-optimal:
 697 accounting for inference in language model scaling laws. In *Proceedings of the 41st International*
698 Conference on Machine Learning, ICML'24. JMLR.org, 2024.

699

700 Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language
 701 models a mirage?, 2023. URL <https://arxiv.org/abs/2304.15004>.

702

703 Noam Shazeer. Glu variants improve transformer, 2020. URL <https://arxiv.org/abs/2002.05202>.

702 Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer:
 703 Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
 704 ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2023.127063>. URL <https://www.sciencedirect.com/science/article/pii/S0925231223011864>.

705
 706 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée
 707 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Ar-
 708 mand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation
 709 language models, 2023. URL <https://arxiv.org/abs/2302.13971>.

710
 711 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,
 712 Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von
 713 Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Ad-
 714 vances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.,
 715 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fdb053c1c4a845aa-Paper.pdf.

716
 717 Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn.
 718 Position: will we run out of data? limits of llm scaling based on human-generated data. In
 719 *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org,
 720 2024.

721
 722 Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani
 723 Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto,
 724 Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large lan-
 725 guage models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL
 726 <https://openreview.net/forum?id=yzkSU5zdwD>. Survey Certification.

727
 728 Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. To repeat or not to
 729 repeat: Insights from scaling llm under token-crisis. *Advances in Neural Information Processing
 Systems*, 36:59304–59322, 2023.

730
 731 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
 732 Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu,
 733 and et al. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.

734
 735 Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws:
 736 Optimizing data mixtures by predicting language modeling performance, 2025. URL <https://arxiv.org/abs/2403.16952>.

737
 738 Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a
 739 machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Márquez
 740 (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*,
 741 pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.
 742 18653/v1/P19-1472. URL <https://aclanthology.org/P19-1472/>.

743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755

756 A TRAINING DATASET
757758 We use the English portion of the Common Crawl Dataset (Common Crawl Foundation), utilizing
759 96 of the snapshots, from CC-MAIN-2013-20 to CC-MAIN-2024-18. Following Bi et al. (2024), we
760 ran a global fuzzy deduplication across all snapshots, resulting in a total dataset with 3.7T tokens.
761762 B LAYERMIX SAMPLING FUNCTION
763764 We show the detail of LayerMix sampling function in Algorithm 1.
765766 **Algorithm 1** LayerMix Sampling Function $H(w, K, S, B)$
767

```

768 1: function  $H(w, K, S, B)$ 
769 Require:
770    $w$ : A list of target proportions for six buckets,  $w = [w_0, \dots, w_5]$ , where  $\sum w_d = 1$ .
771    $K$ : The total number of tokens for the final training dataset.
772    $S$ : The total number of tokens in the entire source corpora.
773    $B$ : The source distribution proportions  $B = [0.05, 0.15, 0.2, 0.2, 0.2, 0.2]$ .
774 Ensure:
775    $D_{train}$ : The final packed training dataset.
776    $M$ : A list of unique token counts for each layer,  $M = [M_0, \dots, M_5]$ .
777    $R$ : A list of average repetition counts for each layer,  $R = [R_0, \dots, R_5]$ .
778
779 2: Initialize an empty training dataset  $D_{train} \leftarrow \emptyset$ .
780 3: Initialize empty lists for statistics:  $M \leftarrow [], R \leftarrow []$ .
781
782 4: for  $d \leftarrow 0$  to 5 do                                 $\triangleright$  Iterate through each quality bucket
783   5:    $K_{needed} \leftarrow K \times w_d$             $\triangleright$  Calculate tokens needed from bucket  $d$  for the target mix
784   6:    $S_d \leftarrow S \times B[d]$             $\triangleright$  Calculate source tokens available in bucket  $d$ 
785   7:    $Ratio_d \leftarrow K_{needed}/S_d$             $\triangleright$  Calculate the sampling ratio for the current bucket
786   8:   Initialize an empty temporary set  $D_{sampled,d} \leftarrow \emptyset$ .
787   9:   for all data point  $x$  in bucket  $d$  do
788     10:       $\triangleright$  1. Deterministic copy for the integer part of the ratio
789     11:      for  $i \leftarrow 1$  to  $\lfloor Ratio_d \rfloor$  do
790       12:        Add  $x$  to  $D_{sampled,d}$ 
791     13:      end for                                 $\triangleright$  2. Probabilistic sampling for the fractional part
792     14:      if  $Ratio_d - \lfloor Ratio_d \rfloor > 0$  and  $random() < (Ratio_d - \lfloor Ratio_d \rfloor)$  then
793       15:        Add  $x$  to  $D_{sampled,d}$ 
794     16:      end if
795     17:      end for
796     18:       $M_d \leftarrow \min(K_{needed}, S_d)$             $\triangleright$  Calculate unique tokens based on the new formula
797     19:      Append  $M_d$  to  $M$ .
798     20:       $R_d \leftarrow K_{needed}/M_d$             $\triangleright$  Calculate average repetition count
799     21:      Append  $R_d$  to  $R$ .
800     22:    end for
801
802 23: return  $D_{train}, M, R$                                  $\triangleright$  Return dataset and statistics
803 24: end function

```

804
805
806
807
808
809
$$\sqrt{m} = \frac{N_{opt}}{N} = \frac{D}{D_{opt}} \quad (10)$$

810 A value of $m = 1$ indicates a compute-optimal training run, while $m > 1$ signifies that the model is
 811 overtrained relative to its compute budget.
 812

Algorithm 2 Calculation of Overtrain Degree and Optimal Tokens

```

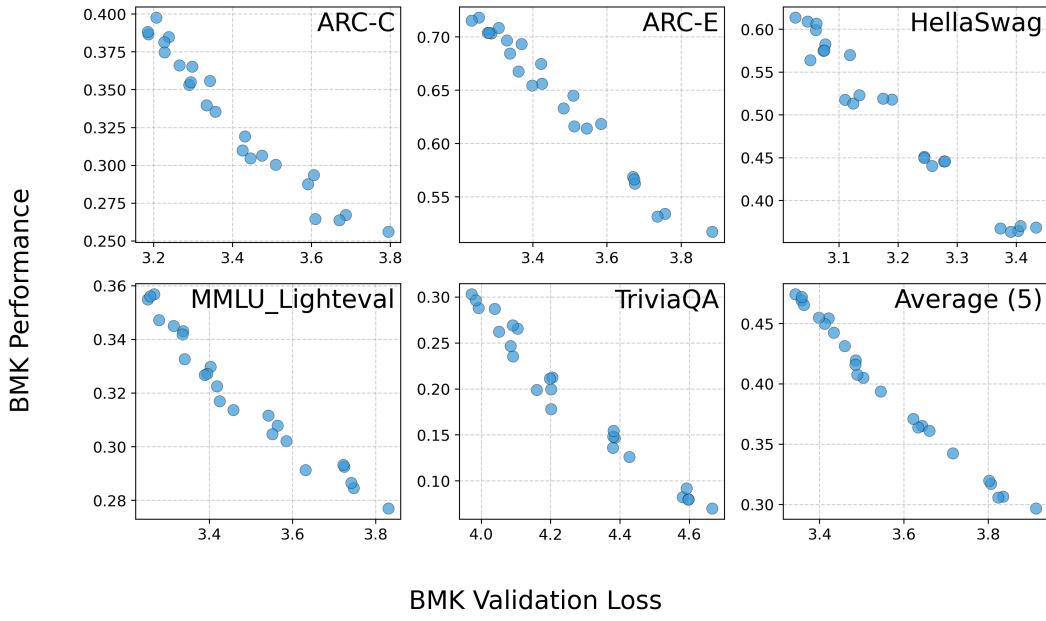
 1: function CALCULATEOVERTRAINEXTRAPOLATION( $model_{curr}$ ,  $D_{curr}$ ,  $models_{target}$ )
 2: Require:
 3:    $model_{curr}$ : The size of the current model configuration.
 4:    $D_{curr}$ : The number of tokens used to train the current model.
 5:    $model_{target}$ : The size of the target model configuration.
 6: Ensure:
 7:    $m$ : The calculated overtrain degree for the current configuration.
 8:    $D_{target}$ : The train token of target model under same overtrain degree.
 9:
10: 2:   // Part 1: Calculate overtrain degree  $m$  from the current configuration
11: 3:    $N_{curr} \leftarrow \text{Get\_N}(model_{curr})$   $\triangleright$  Get  $N$  (non-embedding FLOPs/token) for the current model
12: 4:    $C \leftarrow N_{curr} \times D_{curr}$   $\triangleright$  Calculate the total compute budget
13: 5:    $N_{opt} \leftarrow 0.06085 \times C^{0.5445}$   $\triangleright$  Calculate Chinchilla-optimal model non-embedding
14: 6:    $D_{opt} \leftarrow 16.4326 \times C^{0.4555}$   $\triangleright$  Calculate Chinchilla-optimal tokens for budget  $C$ 
15: 7:    $\sqrt{m} \leftarrow N_{opt}/N_{curr}$   $\triangleright$  Calculate the overtrain degree  $m$ 
16: 8:    $\triangleright$  This is equivalent to  $\sqrt{m} = D_{curr}/D_{opt}$ 
17: 9:   // Part 2: Extrapolate to target model while keeping  $m$  constant
18:10: 10:  for each  $model_t$  in  $[model_{curr}] + models_{target}$  do
19:11: 11:     $N_t \leftarrow \text{Get\_N}(model_t)$   $\triangleright$  Get  $N$  (non-embedding FLOPs/token) for the target model
20:12: 12:     $N'_{opt} \leftarrow N_t \times \sqrt{m}$   $\triangleright$  Find the corresponding optimal model non-embedding
21:13: 13:     $C_{new} \leftarrow (N'_{opt}/0.06085)^{1/0.5445}$   $\triangleright$  Derive the new compute budget
22:14: 14:     $D'_{opt} \leftarrow 16.4326 \times C_{new}^{0.4555}$   $\triangleright$  Find optimal tokens for the new budget
23:15: 15:     $D_{target} \leftarrow D'_{opt} \times \sqrt{m}$   $\triangleright$  Calculate the required tokens for the target model
24:16: 16:  end for
25:
26:17: 17:  return  $m, D_{target}$   $\triangleright$  Return the overtrain degree and the train token of target model under
27:18: 18:  same  $m$ .
28: end function
29:
30:
31: C TRAINING
32:
33: The model structures used in LayerMix are illustrated in Table 4. We train all the model with 2048
34: as the max sequence length, we use a cosine decay scheduler and the initial learning rate calculated
35: by  $lr = \text{round}(0.3118 \cdot C^{-0.1250}, 8)$ , the warm up ratio is set 0.5%. We use AdamW optimizer with
36:  $\beta_1 = 0.9$ ,  $\beta_2 = 0.95$ , weight decay= 0.1.
37:
38: D THE RELATIONSHIP BETWEEN BENCHMARK VALIDATION LOSS AND
39: PERFORMANCE
40:
41: Our LayerMix Law focus on predicting the evaluation loss on downstream benchmarks. However,
42: it also represents for the actual downstream performance. Figure 5 shows a near-linear relationship
43: between validation loss and downstream performance on our evaluation tasks, and Table 5 shows the
44: spearman corelation between validation loss and downstream performance. Lower loss consistently
45:
46:
47:
48:
49:
50:
51:
52:
53:
54:
55:
56:
57:
58:
59:
60:
61:
62:
63:
64:
65:
66:
67:
68:
69:
70:
71:
72:
73:
74:
75:
76:
77:
78:
79:
80:
81:
82:
83:
84:
85:
86:
87:
88:
89:
89:
90:
91:
92:
93:
94:
95:
96:
97:
98:
99:
99:
100:
101:
102:
103:
104:
105:
106:
107:
108:
109:
109:
110:
111:
112:
113:
114:
115:
116:
117:
118:
119:
119:
120:
121:
122:
123:
124:
125:
126:
127:
128:
129:
129:
130:
131:
132:
133:
134:
135:
136:
137:
138:
139:
139:
140:
141:
142:
143:
144:
145:
146:
147:
148:
149:
149:
150:
151:
152:
153:
154:
155:
156:
157:
158:
159:
159:
160:
161:
162:
163:
164:
165:
166:
167:
168:
169:
169:
170:
171:
172:
173:
174:
175:
176:
177:
178:
179:
179:
180:
181:
182:
183:
184:
185:
186:
187:
188:
189:
189:
190:
191:
192:
193:
194:
195:
196:
197:
198:
199:
199:
200:
201:
202:
203:
204:
205:
206:
207:
208:
209:
209:
210:
211:
212:
213:
214:
215:
216:
217:
217:
218:
219:
219:
220:
221:
222:
223:
224:
225:
226:
227:
227:
228:
229:
229:
230:
231:
232:
233:
234:
235:
236:
236:
237:
238:
238:
239:
239:
240:
241:
242:
243:
244:
245:
245:
246:
247:
247:
248:
248:
249:
249:
250:
251:
252:
253:
253:
254:
255:
255:
256:
256:
257:
257:
258:
258:
259:
259:
260:
260:
261:
261:
262:
262:
263:
263:
264:
264:
265:
265:
266:
266:
267:
267:
268:
268:
269:
269:
270:
270:
271:
271:
272:
272:
273:
273:
274:
274:
275:
275:
276:
276:
277:
277:
278:
278:
279:
279:
280:
280:
281:
281:
282:
282:
283:
283:
284:
284:
285:
285:
286:
286:
287:
287:
288:
288:
289:
289:
290:
290:
291:
291:
292:
292:
293:
293:
294:
294:
295:
295:
296:
296:
297:
297:
298:
298:
299:
299:
300:
300:
301:
301:
302:
302:
303:
303:
304:
304:
305:
305:
306:
306:
307:
307:
308:
308:
309:
309:
310:
310:
311:
311:
312:
312:
313:
313:
314:
314:
315:
315:
316:
316:
317:
317:
318:
318:
319:
319:
320:
320:
321:
321:
322:
322:
323:
323:
324:
324:
325:
325:
326:
326:
327:
327:
328:
328:
329:
329:
330:
330:
331:
331:
332:
332:
333:
333:
334:
334:
335:
335:
336:
336:
337:
337:
338:
338:
339:
339:
340:
340:
341:
341:
342:
342:
343:
343:
344:
344:
345:
345:
346:
346:
347:
347:
348:
348:
349:
349:
350:
350:
351:
351:
352:
352:
353:
353:
354:
354:
355:
355:
356:
356:
357:
357:
358:
358:
359:
359:
360:
360:
361:
361:
362:
362:
363:
363:
364:
364:
365:
365:
366:
366:
367:
367:
368:
368:
369:
369:
370:
370:
371:
371:
372:
372:
373:
373:
374:
374:
375:
375:
376:
376:
377:
377:
378:
378:
379:
379:
380:
380:
381:
381:
382:
382:
383:
383:
384:
384:
385:
385:
386:
386:
387:
387:
388:
388:
389:
389:
390:
390:
391:
391:
392:
392:
393:
393:
394:
394:
395:
395:
396:
396:
397:
397:
398:
398:
399:
399:
400:
400:
401:
401:
402:
402:
403:
403:
404:
404:
405:
405:
406:
406:
407:
407:
408:
408:
409:
409:
410:
410:
411:
411:
412:
412:
413:
413:
414:
414:
415:
415:
416:
416:
417:
417:
418:
418:
419:
419:
420:
420:
421:
421:
422:
422:
423:
423:
424:
424:
425:
425:
426:
426:
427:
427:
428:
428:
429:
429:
430:
430:
431:
431:
432:
432:
433:
433:
434:
434:
435:
435:
436:
436:
437:
437:
438:
438:
439:
439:
440:
440:
441:
441:
442:
442:
443:
443:
444:
444:
445:
445:
446:
446:
447:
447:
448:
448:
449:
449:
450:
450:
451:
451:
452:
452:
453:
453:
454:
454:
455:
455:
456:
456:
457:
457:
458:
458:
459:
459:
460:
460:
461:
461:
462:
462:
463:
463:
464:
464:
465:
465:
466:
466:
467:
467:
468:
468:
469:
469:
470:
470:
471:
471:
472:
472:
473:
473:
474:
474:
475:
475:
476:
476:
477:
477:
478:
478:
479:
479:
480:
480:
481:
481:
482:
482:
483:
483:
484:
484:
485:
485:
486:
486:
487:
487:
488:
488:
489:
489:
490:
490:
491:
491:
492:
492:
493:
493:
494:
494:
495:
495:
496:
496:
497:
497:
498:
498:
499:
499:
500:
500:
501:
501:
502:
502:
503:
503:
504:
504:
505:
505:
506:
506:
507:
507:
508:
508:
509:
509:
510:
510:
511:
511:
512:
512:
513:
513:
514:
514:
515:
515:
516:
516:
517:
517:
518:
518:
519:
519:
520:
520:
521:
521:
522:
522:
523:
523:
524:
524:
525:
525:
526:
526:
527:
527:
528:
528:
529:
529:
530:
530:
531:
531:
532:
532:
533:
533:
534:
534:
535:
535:
536:
536:
537:
537:
538:
538:
539:
539:
540:
540:
541:
541:
542:
542:
543:
543:
544:
544:
545:
545:
546:
546:
547:
547:
548:
548:
549:
549:
550:
550:
551:
551:
552:
552:
553:
553:
554:
554:
555:
555:
556:
556:
557:
557:
558:
558:
559:
559:
560:
560:
561:
561:
562:
562:
563:
563:
564:
564:
565:
565:
566:
566:
567:
567:
568:
568:
569:
569:
570:
570:
571:
571:
572:
572:
573:
573:
574:
574:
575:
575:
576:
576:
577:
577:
578:
578:
579:
579:
580:
580:
581:
581:
582:
582:
583:
583:
584:
584:
585:
585:
586:
586:
587:
587:
588:
588:
589:
589:
590:
590:
591:
591:
592:
592:
593:
593:
594:
594:
595:
595:
596:
596:
597:
597:
598:
598:
599:
599:
600:
600:
601:
601:
602:
602:
603:
603:
604:
604:
605:
605:
606:
606:
607:
607:
608:
608:
609:
609:
610:
610:
611:
611:
612:
612:
613:
613:
614:
614:
615:
615:
616:
616:
617:
617:
618:
618:
619:
619:
620:
620:
621:
621:
622:
622:
623:
623:
624:
624:
625:
625:
626:
626:
627:
627:
628:
628:
629:
629:
630:
630:
631:
631:
632:
632:
633:
633:
634:
634:
635:
635:
636:
636:
637:
637:
638:
638:
639:
639:
640:
640:
641:
641:
642:
642:
643:
643:
644:
644:
645:
645:
646:
646:
647:
647:
648:
648:
649:
649:
650:
650:
651:
651:
652:
652:
653:
653:
654:
654:
655:
655:
656:
656:
657:
657:
658:
658:
659:
659:
660:
660:
661:
661:
662:
662:
663:
663:
664:
664:
665:
665:
666:
666:
667:
667:
668:
668:
669:
669:
670:
670:
671:
671:
672:
672:
673:
673:
674:
674:
675:
675:
676:
676:
677:
677:
678:
678:
679:
679:
680:
680:
681:
681:
682:
682:
683:
683:
684:
684:
685:
685:
686:
686:
687:
687:
688:
688:
689:
689:
690:
690:
691:
691:
692:
692:
693:
693:
694:
694:
695:
695:
696:
696:
697:
697:
698:
698:
699:
699:
700:
700:
701:
701:
702:
702:
703:
703:
704:
704:
705:
705:
706:
706:
707:
707:
708:
708:
709:
709:
710:
710:
711:
711:
712:
712:
713:
713:
714:
714:
715:
715:
716:
716:
717:
717:
718:
718:
719:
719:
720:
720:
721:
721:
722:
722:
723:
723:
724:
724:
725:
725:
726:
726:
727:
727:
728:
728:
729:
729:
730:
730:
731:
731:
732:
732:
733:
733:
734:
734:
735:
735:
736:
736:
737:
737:
738:
738:
739:
739:
740:
740:
741:
741:
742:
742:
743:
743:
744:
744:
745:
745:
746:
746:
747:
747:
748:
748:
749:
749:
750:
750:
751:
751:
752:
752:
753:
753:
754:
754:
755:
755:
756:
756:
757:
757:
758:
758:
759:
759:
760:
760:
761:
761:
762:
762:
763:
763:
764:
764:
765:
765:
766:
766:
767:
767:
768:
768:
769:
769:
770:
770:
771:
771:
772:
772:
773:
773:
774:
774:
775:
775:
776:
776:
777:
777:
778:
778:
779:
779:
780:
780:
781:
781:
782:
782:
783:
783:
784:
784:
785:
785:
786:
786:
787:
787:
788:
788:
789:
789:
790:
790:
791:
791:
792:
792:
793:
793:
794:
794:
795:
795:
796:
796:
797:
797:
798:
798:
799:
799:
800:
800:
801:
801:
802:
802:
803:
803:
804:
804:
805:
805:
806:
806:
807:
807:
808:
808:
809:
809:
810:
810:
811:
811:
812:
812:
813:
813:
814:
814:
815:
815:
816:
816:
817:
817:
818:
818:
819:
819:
820:
820:
821:
821:
822:
822:
823:
823:
824:
824:
825:
825:
826:
826:
827:
827:
828:
828:
829:
829:
830:
830:
831:
831:
832:
832:
833:
833:
834:
834:
835:
835:
836:
836:
837:
837:
838:
838:
839:
839:
840:
840:
841:
841:
842:
842:
843:
843:
844:
844:
845:
845:
846:
846:
847:
847:
848:
848:
849:
849:
850:
850:
851:
851:
852:
852:
853:
853:
854:
854:
855:
855:
856:
856:
857:
857:
858:
858:
859:
859:
860:
860:
861:
861:
862:
862:
863:
863:
864:
864:
865:
865:
866:
866:
867:
867:
868:
868:
869:
869:
870:
870:
871:
871:
872:
872:
873:
873:
874:
874:
875:
875:
876:
876:
877:
877:
878:
878:
879:
879:
880:
880:
881:
881:
882:
882:
883:
883:
884:
884:
885:
885:
886:
886:
887:
887:
888:
888:
889:
889:
890:
890:
891:
891:
892:
892:
893:
893:
894:
894:
895:
895:
896:
896:
897:
897:
898:
898:
899:
899:
900:
900:
901:
901:
902:
902:
903:
903:
904:
904:
905:
905:
906:
906:
907:
907:
908:
908:
909:
909:
910:
910:
911:
911:
912:
912:
913:
913:
914:
914:
915:
915:
916:
916:
917:
917:
918:
918:
919:
919:
920:
920:
921:
921:
922:
922:
923:
923:
924:
924:
925:
925:
926:
926:
927:
927:
928:
928:
929:
929:
930:
930:
931:
931:
932:
932:
933:
933:
934:
934:
935:
935:
936:
936:
937:
937:
938:
938:
939:
939:
940:
940:
941:
941:
942:
942:
943:
943:
944:
944:
945:
945:
946:
946:
947:
947:
948:
948:
949:
949:
950:
950:
951:
951:
952:
952:
953:
953:
954:
954:
955:
955:
956:
956:
957:
957:
958:
958:
959:
959:
960:
960:
961:
961:
962:
962:
963:
963:
964:
964:
965:
965:
966:
966:
967:
967:
968:
968:
969:
969:
970:
970:
971:
971:
972:
972:
973:
973:
974:
974:
975:
975:
976:
976:
977:
977:
978:
978:
979:
979:
980:
980:
981:
981:
982:
982:
983:
983:
984:
984:
985:
985:
986:
986:
987:
987:
988:
988:
989:
989:
990:
990:
991:
991:
992:
992:
993:
993:
994:
994:
995:
995:
996:
996:
997:
997:
998:
998:
999:
999:
1000:
1000:
1001:
1001:
1002:
1002:
1003:
1003:
1004:
1004:
1005:
1005:
1006:
1006:
1007:
1007:
1008:
1008:
1009:
1009:
1010:
1010:
1011:
1011:
1012:
1012:
1013:
1013:
1014:
1014:
1015:
1015:
1016:
1016:
1017:
1017:
1018:
1018:
1019:
1019:
1020:
1020:
1021:
1021:
1022:
1022:
1023:
1023:
1024:
1024:
1025:
1025:
1026:
1026:
1027:
1027:
1028:
1028:
1029:
1029:
1030:
1030:
1031:
1031:
1032:
1032:
1033:
1033:
1034:
1034:
1035:
1035:
1036:
1036:
1037:
1037:
1038:
1038:
1039:
1039:
1040:
1040:
1041:
1041:
1042:
1042:
1043:
1043:
1044:
1044:
1045:
1045:
1046:
1046:
1047:
1047:
1048:
1048:
1049:
1049:
1050:
1050:
1051:
1051:
1052:
1052:
1053:
1053:
1054:
1054:
1055:
1055:
1056:
1056:
1057:
1057:
1058:
1058:
1059:
1059:
1060:
1060:
1061:
1061:
1062:
1062:
1063:
1063:
1064:
1064:
1065:
1065:
1066:
1066:
1067:
1067:
1068:
1068:
1069:
1069:
1070:
1070:
1071:
1071:
1072:
1072:
1073:
1073:
1074:
1074:
1075:
1075:
1076:
1076:
1077:
1077:
1078:
1078:
1079:
1079:
1080:
1080:
1081:
1081:
1082:
1082:
1083:
1083:
1084:
1084:
1085:
1085:
1086:
1086:
1087:
1087:
1088:
1088:
1089:
1089:
1090:
1090:
1091:
1091:
1092:
1092:
1093:
1093:
1094:
1094:
1095:
1095:
1096:
1096:
1097:
1097:
1098:
1098:
1099:
1099:
1100:
1100:
1101:
1101:
1102:
1102:
1103:
1103:
1104:
1104:
1105:
1105:
1106:
1106:
1107:
1107:
1108:
1108:
1109:
1109:
1110:
1110:
1111:
1111:
1112:
1112:
1113:
1113:
1114:
1114:
1115:
1115:
1116:
1116:
1117:
1117:
1118:
1118:
1119:
1119:
1120:
1120:
1121:
1121:
1122:
1122:
1123:
1123:
1124:
1124:
1125:
1125:
1126:
1126:
1127:
1127:
1128:
1128:
1129:
1129:
1130:
1130:
1131:
1131:
1132:
1132:
1133:
1133:
1134:
1134:
1135:
1135:
1136:
1136:
1137:
1137:
1138:
1138:
1139:
1139:
1140:
1140:
1141:
1141:
1142:
1142:
1143:
1143:
1144:
1144:
1145:
1145:
1146:
1146:
1147:
1147:
1148:
1148:
1149:
1149:
1150:
1150:
1151:
1151:
1152:
1152:
1153:
1153:
1154:
1154:
1155:
1155:
1156:
1156:
1157:
1157:
1158:
1158:
1159:
1159:
1160:
1160:
1161:
1161:
1162:
1162:
1163:
1163:
1164:
1164:
1165:
1165:
1166:
1166:
1167:
1167:
1168:
1168:
1169:
1169:
1170:
1170:
1171:
1171:
1172:
1172:
1173:
1173:
1174:
1174:
1175:
1175:
1176:
1176:
1177:
1177:
1178:
1178:
1179:
1179:
1180:
1180:
1181:
1181:
1182:
1182:
1183:
1183:
1184:
1184:
1185:
1185:
1186:
1186:
1187:
1187:
1188:
1188:
1189:
1189:
1190:
1190:
1191:
1191:
1192:
1192:
1193:
1193:
1194:
1194:
1195:
1195:
1196:
1196:
1197:
1197:
1198:
1198:
1199:
1199:
1200:
1200:
1201:
1201:
1202:
1202:
1203:
1203:
1204:
1204:
1205:
1205:
1206:
1206:
1207:
1207:
1208:
1208:
1209:
1209:
1210:
1210:
1211:
1211:
1212:
1212:
1213:
1213:
1214:
1214:
1215:
1215:
1216:
1216:
1217:
1217:
1218:
1218:
1219:
1219:
1220:
1220:
1221:
1221:
1222:
1222:
1223:
1223:
1224:
1224:
1225:
1225:
1226:
1226:
1227:
1227:
1228:
1228:
1229:
1229:
1230:
1230:
1231:
1231:
1232:
1232:
1233:
1233:
1234:
1234:
1235:
1235:
1236:
1236:
1237:
1237:
1238:
1238:
1239:
1239:
1240:
1240:
1241:
1241:
1242:
1242:
1243:
1243:
1244:
1244:
1245:
1245:
1246:
1246:
1247:
1247:
1248:
1248:
1249:
1249:
1250:
1250:
1251:
1251:
1252:
1252:
1253:
1253:
1254:
1254:
1255:
1255:
1256:
1256:
1257:
1257:
1258:
1258:
1259:
1259:
1260:
1260:
1261:
1261:
1262:
1262:
1263:
1263:
1264:
1264:
1265:
1265:
1266:
1266:
1267:
1267:
1268:
1268:
1269:
1269:
1270:
1270:
1271:
1271:
1272:
1272:
1273:
1273:
1274:
1274:
1275:
1275:
1276:
1276:
1277:
1277:
1278:
1278:
1279:
1279:
1280:
1280:
1281:
1281:
1282:
1282:
1283:
1283:
1284:
1284:
1285:
1285:
1286:
1286:
1287:
1287:
1288:
1288:
1289:
1289:
1290:
1290:
1291:
1291:
1292:
1292:
1293:
1293:
1294:
1294:
1295:
1295:
1296:
1296:
1297:
1297:
1298:
1298:
1299:
1299:
1300:
1300:
1301:
1301:
1302:
1302:
1303:
1303:
1304:
1304:
1305:
1305:
1306:
1306:
1307
```

864
865

Table 4: Structure of models used in LayerMix.

866
867
868
869
870
871
872
873
874
875
876
877
878

Model	Hidden dim. (C)	MLP dim. (D)	Layers (L)	Heads
252M	1024	2752	20	16
302M	1024	2752	24	16
392M	1280	3392	20	20
470M	1280	3392	24	20
566M	1536	4096	20	24
680M	1536	4096	24	24
850M	1792	4800	22	28
1B	1920	5120	24	30
1.2B	2048	5440	24	16
1.5B	2304	6144	24	36
1.8B	2304	6144	28	36
2.5B	2560	6848	32	40
7.7B	4096	14336	32	32

879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899

900

901 Figure 5: Validation loss versus downstream performance across benchmarks (ARC-C, ARC-E,
902 HellaSwag, MMLU-Lighteval, TriviaQA) and their average.903
904
905

906 corresponds to higher performance within the operating regime of our models. This indicates that
907 improvements in loss provide reliable signals for expected gains in downstream performance.

908
909
910
911

E ALTERNATIVE FITS FOR λ

912
913
914
915

In Section 5.2, we model the relationship between non-embedding FLOPs/token N and hyperparameter λ . Our primary specification adopts the logarithmic form Equation 9. Beyond this baseline, we also evaluated alternative function families, including an exponential form:

$$\lambda(x; a, b, c) = a \cdot (1 - e^{-bx+c}) \quad (11)$$

and a power-law form:

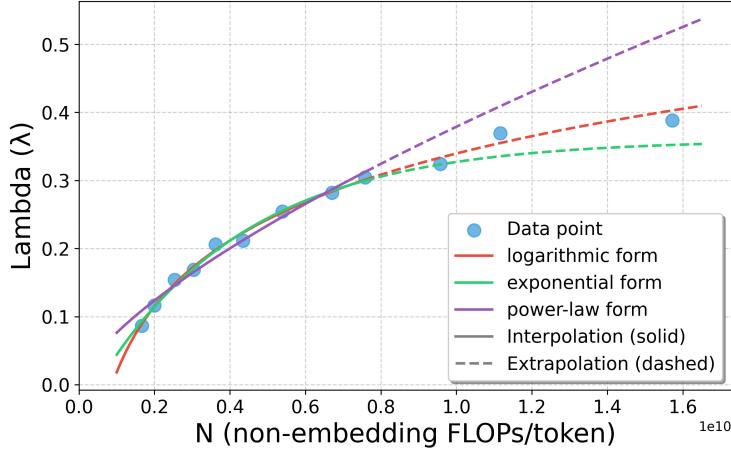
$$\lambda(x; a, b) = a \cdot x^b \quad (12)$$

916
917

As shown in Figure 6, the logarithmic model achieves the best fit to the $N - \lambda$ relationship, outperforming the exponential and power-law alternatives. Accordingly, we adopt function 9 as the final parameterization.

918 Table 5: Spearman correlation between validation loss and performance across benchmarks
919

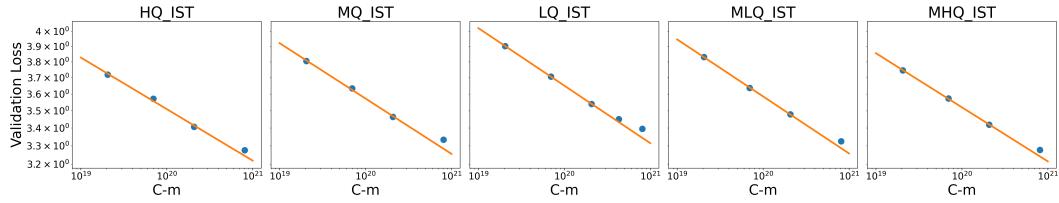
Benchmark	Spearman r_s	p -value
ARC-C	-0.979	1.02×10^{-16}
ARC-E	-0.982	2.72×10^{-17}
HellaSwag	-0.942	6.13×10^{-12}
MMLU-LightEval	-0.989	1.26×10^{-19}
TriviaQA	-0.970	4.53×10^{-15}
Average (5)	-0.996	3.54×10^{-24}

931 Figure 6: Comparison of functional fits for λ as a function of N (non-embedding FLOPs/token).
932 The logarithmic form provides the best in-domain fit and extrapolation behavior compared with the
933 exponential and power-law alternatives. Solid lines denote interpolation over observed N ; dashed
934 lines indicate extrapolation beyond the observed range.
935936

F DEVIATION OF TRADITIONAL SCALING LAW

937 We show all $Loss-C$ curve of different LayerMix sampling weights with IST and LST in Figure 7
938 and Figure 8, they all exhibit a clear deviation from the traditional scaling law, which is fitted from
939 the first three data points.
940941

G QUALITY SCORE

942 We show some data samples in different Quality buckets in Figure 9. This figure indicates that high-
943 score samples under our merged FineWebEdu and DCLM scores are more coherent and instruc-
944 tional. By contrast, low-score cases predominantly consist of advertisements or low-information
945 content, offering little substantive value.
946947 Table 6 reports four benchmark results for training a 1.2B model from scratch on 30B tokens using
948 three datasets: the top 5% and top 20% selected by the FineWebEdu classifier, and a random sample,
949950 Figure 7: $Loss$ and C_m Curve of different LayerMix IST experiments
951

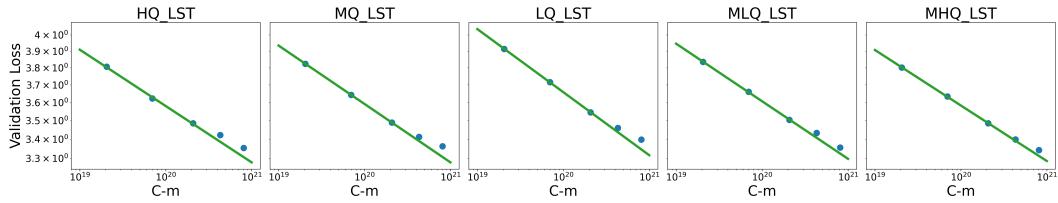
Figure 8: *Loss and C_m Curve of different LayerMix LST experiments*

Table 6: FineWebEdu-selected subsets vs. random data for training a 1.2B model on 30B tokens

Model	Data	ARC-C	HellaSwag	TriviaQA	MMLU-LightEval	avg
1.2B	Random 30B	28.50%	51.56%	15.55%	30.23%	31.46%
1.2B	FWE-top20% 30B	34.30%	55.26%	20.05%	32.82%	35.61%
1.2B	FWE-top5% 30B	37.20%	55.14%	19.25%	34.50%	36.52%

all from Penedo et al. (2023). High-quality data selected by FineWebEdu outperforms the random baseline, and higher-quality subsets yield better results.

H OPTIMIZING TOKEN MIX WITH LAYERMIX LAW

We present the detailed optimal LayerMix parameters (or token-mix ratios) for different models and training budgets predicted by LayerMix Law in Table 7. This table shows that small models or small training budgets prioritize quality, while large models or large training budgets prioritize diversity.

I GENERALIZATION TO REFINEDWEB

To evaluate the robustness and generalization capability of the LayerMix Law across different data distributions, we conducted an additional series of verification experiments on the RefinedWeb dataset (Penedo et al., 2023).

Experimental Setup. We followed the identical data preprocessing, LayerMix sampling, and training procedures described in Section 3.1 and Section 5.1, with the sole exception of replacing the source corpus with RefinedWeb. Due to time and computational constraints, we limited the scope of this study to three LayerMix sampling configurations: HQ (High Quality) and LQ (Low Quality) were used for parameter fitting (interpolation), while MLQ (Medium-Low Quality) was held out for extrapolation testing. For each configuration, we trained models at three specific scales: 302M, 566M, and 1.2B parameters.

Fitting and Extrapolation. We applied the fitting methodology outlined in Section 5.2. Our analysis yielded two key observations:

- **Consistency of Quality Density (f):** The fitted values for the quality density function f_d were numerically very close to those derived from our primary dataset. Specifically, the fitted parameter θ is 0.93 for RefinedWeb, which is remarkably close to the value of 0.92 obtained from our primary dataset. We attribute this similarity to the fact that RefinedWeb (Penedo et al., 2023) is also derived from Common Crawl (Common Crawl Foundation); despite employing different filtering strategies, the shared underlying data source results in a comparable information density distribution.
- **Optimization of λ_N :** In the main experiments, we modeled the relationship between the parameter λ_N and model scale N using a logarithmic curve. However, due to the limited number of data points in this verification set (only three distinct model scales), fitting a robust $\lambda_N - N$ curve was not feasible. Consequently, we skipped the curve fitting step for λ_N and directly searched for the optimal λ values corresponding to the specific model sizes (302M, 566M, and 1.2B).

1026
1027
1028
1029
1030
1031

1032
1033

Quality_Range 0-5%	Quality_Range 80-100%
<p>Celebrate your way</p> <p>Whether you are having a picnic with your family, a barbecue with friends in the backyard or follow the Australian of the Year awards, Australia Day on the 26th of January is an occasion to come together as a nation and celebrate what's great about Australia.</p> <p>On Australia Day we celebrate the past, present and future of the country. It is a commemoration of the day that the First Fleet landed in Sydney Cove in 1788, as well as a celebration of all the achievements of our country.</p> <p>The tradition of having Australia Day as a national holiday on 26 January is actually a pretty recent one. Not until 1935 did all Australian states and territories use that name to mark that date, and only in 1994 did they begin to celebrate Australia Day as a public holiday on that date.</p> <p>Today, Australia Day has grown to be a community day which is embraced by most Australians. Apart from the formal ceremonies around the country such as flag raising, citizenship ceremonies and the presentation of community awards, there are a wider range of festivities that encourage the participation of all family and community members.</p>	<p>This message was posted by The Dumper, posted on January 05, 2002 at 03:06:18 coming from 209.204.139 This message is a reply to Will BUY snes copier posted from Spongebob posted at January 05, 2002 at 02:25:29> Looking for an snes copier. Preferably the Super Wildcard DX 2. Will pay \$\$\$\$\$ or it.</p>
<p>March 7, 2012 (Shirley Allen)</p> <p>Monthly home prices in the United States increased by a seasonally adjusted 0.7 percent in December which follows a similarly revised 0.7 percent gain in November according to the Federal Housing Finance Agency's (FHFA) monthly House Price Index (HPI).</p> <p>December's home prices were still 0.8 percent lower than they were a year ago and since the market peak in April 2007, home prices have declined over 18 percent and are at roughly the same levels last seen in March of 2004.</p> <p>Six of the nine Census Divisions posted monthly price gains in December with the Mountain Division recording the most improvement of 2.5 percent. Two Divisions posted declines in home prices while one Division, the Middle Atlantic, remained unchanged from the previous month. Of the two Divisions that posted declines, the West North Central Division posted the largest decline of 0.9 percent.</p> <p>Seven of the Divisions registered year-over-year price declines with the Pacific Division posting the largest decline of 3.8 percent. The only two Divisions that posted an increase in annual home prices were the East South Central Division and the West South Central Division which posted increases of 3.0 and 1.7 percent, respectively.</p>	<p>these adorable little wall hangings feature bright pops of colour, tassley texture and pretty little berry knots. How can you resist?>Plus...you get to choose from our gorge range of colours>Made from 100% recycled cotton and mounted on a Tasmanian Oak dowel.</p> <p>Measures approximately 16cm wide by 33cm long (including hanger).>This item is hand made with love and ready to ship. Colours vary from screen to screen.</p> <p>>Looking for something similar? Custom orders are available – price available upon request.</p> <p>*Outside Australia? Please contact us for country specific freight charges.'</p>

1071
1072
1073
1074

Figure 9: Case study contrasting data quality. Left (0–5% quality range): coherent, informational, and instructional passages. Right (80–100% quality range): low-information, ad-like content with minimal reasoning or educational value.

1075
1076
1077
1078
1079

1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1090
 1091
 1092
 1093
 1094

Table 7: The detailed best layer token mix for different models and train token

Model	Train Token	Source Token	w1	w2	w3	w4	w5	w6
7B	200B	500B	0.619	0.376	0.004	0.001	0.000	0.000
	300B	500B	0.548	0.444	0.004	0.003	0.002	0.000
	400B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	500B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	600B	500B	0.491	0.487	0.017	0.005	0.000	0.000
	700B	500B	0.439	0.430	0.130	0.001	0.000	0.000
	800B	500B	0.439	0.430	0.130	0.001	0.000	0.000
	900B	500B	0.404	0.403	0.183	0.006	0.003	0.000
	1000B	500B	0.395	0.387	0.214	0.003	0.001	0.000
	200B	500B	0.825	0.165	0.005	0.004	0.001	0.000
1.8B	300B	500B	0.619	0.376	0.004	0.001	0.000	0.000
	400B	500B	0.548	0.444	0.004	0.003	0.002	0.000
	500B	500B	0.548	0.444	0.004	0.003	0.002	0.000
	600B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	700B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	800B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	900B	500B	0.491	0.487	0.017	0.005	0.000	0.000
	1000B	500B	0.491	0.487	0.017	0.005	0.000	0.000
	200B	500B	0.926	0.066	0.006	0.002	0.000	0.000
	300B	500B	0.758	0.229	0.012	0.001	0.000	0.000
1.2B	400B	500B	0.619	0.376	0.004	0.001	0.000	0.000
	500B	500B	0.619	0.376	0.004	0.001	0.000	0.000
	600B	500B	0.548	0.444	0.004	0.003	0.002	0.000
	700B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	800B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	900B	500B	0.496	0.492	0.007	0.003	0.002	0.000
	1000B	500B	0.496	0.492	0.007	0.003	0.002	0.000

1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133

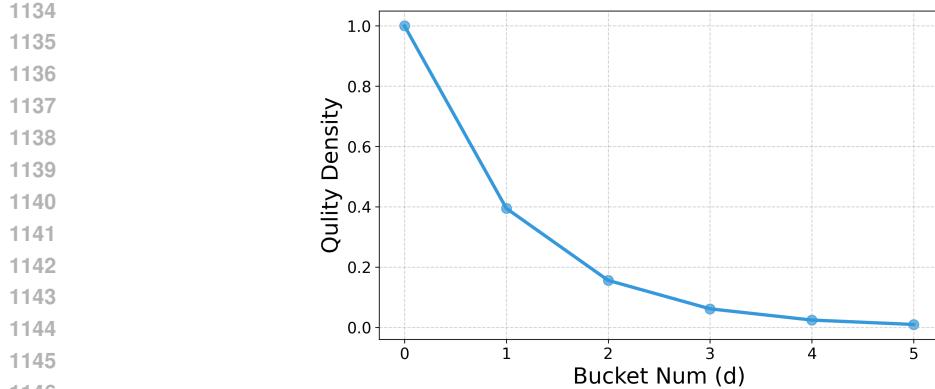
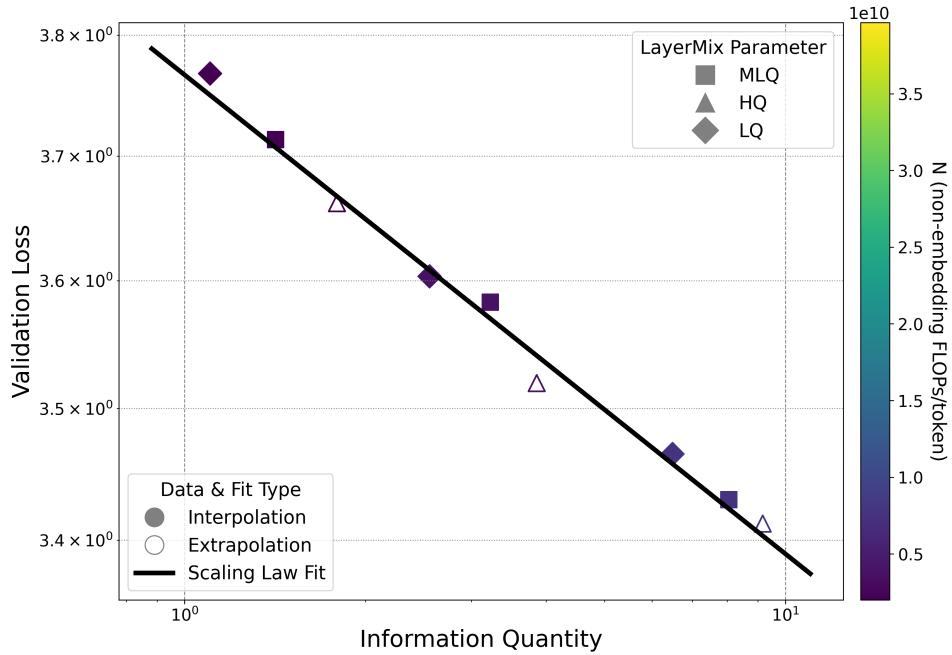
Figure 10: The fitted quality density function f_d on the RefinedWeb dataset.

Figure 11: The Unified Information-Loss Scaling Law on the RefinedWeb dataset.

1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828

1188 **K USAGE OF LLM**
11891190 During the preparation of this paper, large language models (LLMs) were utilized. The use of these
1191 tools was only for polishing the language, improving grammatical structure, and performing spell
1192 checks.
11931194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241