Exploiting Vocabulary Frequency Imbalance in Language Model Pre-training

Woojin Chung*
KAIST†
gartland223@gmail.com

Jeonghoon Kim*
NAVER Cloud & KAIST
jeonghoon.samuel@gmail.com

Abstract

Large language models are trained with tokenizers, and the resulting token distribution is highly *imbalanced*: a few words dominate the stream while most occur rarely. Recent practice favors ever-larger vocabularies, but it is unclear where the benefit comes from. To this end, we perform a controlled study that scales the vocabulary of the language model from 24K to 196K while holding data, computation, and optimization unchanged. We begin by quantifying the complexity of tokenized text – formalized via Kolmogorov complexity – and show that larger vocabularies reduce this complexity. Above 24K, every common word is already tokenized as a single token, so enlarging vocabulary only deepens the relative token-frequency imbalance. Word-level loss decomposition shows that larger vocabularies reduce cross-entropy loss almost exclusively by lowering uncertainty on the $2,500$ most frequent words, even though loss on the rare tail rises. Constraining input and output embedding norms to attenuate the impact of token-frequency imbalance reverses the gain, directly demonstrating that the model *exploits* rather than suffers from imbalance. Because the same frequent words cover roughly 77% of tokens in downstream benchmarks, this training advantage transfers intact. We further show that enlarging model parameters with a fixed vocabulary yields the same frequent-word benefit. Our results recast "bigger vocabularies help" as "lowering complexity of tokenized text helps," offering a simple, principled knob for tokenizer–model co-design and clarifying the loss dynamics that govern language model scaling in pre-training.

1 Introduction

A language model incorporates a tokenizer that converts a stream of characters into a series of token IDs, each representing a specific substring [20, 40, 12]. Tokenization has re-emerged as an powerful tuning knob for language models, with mounting evidence that simply scaling up vocabulary consistently reduces perplexity and improves downstream accuracy across diverse domains and model scales [50, 54]. Although this trend is consistently observed in practice, the underlying mechanism responsible for it has yet to be thoroughly investigated.

As vocabulary size grows, adding merge candidates segments frequent words in the training data into a single token and pushes infrequent ones deeper into the long tail, sharpening the relative token-frequency distribution [44, 28, 19]. In unigram models, this simply lowers the Shannon entropy of training data toward its optimal loss (i.e., entropy rate) [40]. This intuition does not transfer to language models as they condition their next-token prediction on the preceding context, which contains a mixture of common and rare tokens [2, 8, 35, 7]. Moreover, rare tokens already carry

*Equal contribution
†Korea Advanced Institute of Science and Technology (KAIST)

Preprint. Under review.

much lower conditional probabilities than marginal ones; mistakes on them incur disproportionately large penalties [38, 32]. Yet rare tokens account for a small share of the entire dataset, making it unclear how a larger vocabulary reallocates capacity between frequent and rare tokens.

In this study, we explore why enlarging vocabulary size improves the performance of language models by expanding vocabulary from 24K to 196K. We quantify the complexity of tokenized text using an upper bound on Kolmogorov complexity and first demonstrate that expanding the vocabulary reduces this complexity (§3.3). Above 24K, frequent words are already encoded as single tokens, so the primary shift is a heightened imbalance in relative token-frequency distribution (§3.4), regardless of dataset quality (§3.5). Furthermore, a word-level loss decomposition shows that a larger vocabulary reduces the loss of frequent words, thereby lowering the model's overall cross-entropy. (§3.6).

Analytic experiments trace how enlarging the vocabulary changes both training dynamics and generalization behavior through token-frequency imbalance. Our observation suggests that exploiting relative token-frequency imbalance during pre-training is essential for performance (§4.1). In addition, high frequency words in the pre-training corpus largely coincide with those in downstream benchmarks, both in identity and coverage, explaining the close link between training loss and transfer accuracy (§4.2). Scaling the model itself produces the same benefit: it predicts frequent words more accurately, thereby enhancing overall language-model performance (§4.3).

Contribution. We identify that larger vocabularies reduce the complexity of tokenized text, thereby facilitating the model to learn non-i.i.d. patterns in the training data more easily. Our experiments further reveal that beyond a certain size, vocabulary expansion no longer improves segmentation but instead steepens the skewness of the token-frequency distribution. This sharper imbalance alone lowers global cross-entropy by reducing the top $\sim 2{,}500$ frequent words loss despite slight degradation on the rare tail. Through norm-constrained ablations and cross-dataset overlap analyses, we demonstrate that exploiting, rather than mitigating, frequency skew causally reduces cross-entropy and boosts downstream accuracy. Finally, we show that parameter scaling replicates the same benefit as vocabulary scaling, both primarily reduce uncertainty on the same set of frequent tokens.

2 Motivation

Tokenization – the interface between raw text and the discrete symbols a model actually sees – has resurfaced as a powerful, low-cost lever for improving language model quality [20, 40, 12, 19]. A growing body of evidence finds that simply *increasing the size of the tokenizer vocabulary* yields systematic gains in perplexity and downstream accuracy across domain and model scales [50, 54]. Despite this empirical regularity, the mechanism behind the gain remains underexplored.

The clue may lie in how tokenizers behave as their vocabularies grow. Rajaraman et al. [40] noted that adding merge candidates segments the most frequent words in the corpus into single tokens, making an i.i.d. model over tokens a closer approximation to inherently non-i.i.d. data. Simultaneously, this pushes rare vocabularies further into the long tail, yielding a markedly more skewed token-frequency distribution (after the usual whitespace pre-tokenization) [44, 28]. In a unigram model, increasing the relative frequency of common vocabularies reduces Shannon entropy – the theoretical minimum cross-entropy – while a longer tail raises it [56]. Empirically, Rajaraman et al. [40] confirms that larger vocabularies lower unigram cross-entropy.

Unfortunately, this explanation does not carry over verbatim to neural language models. In language models, every prediction is conditioned on a variable-length context. Errors on rare vocabularies are disproportionately costly because their conditional probabilities are already orders of magnitude below their marginal frequencies [38, 32], yet they occupy only a tiny fraction of the overall loss. Quantifying how vocabulary growth redistributes modelling capacity between these two regimes, therefore, remains a non-trivial challenge (see the Appendix A for a detailed explanation).

In this work, we tackle the problem head-on by pairing large-scale controlled experiments with analytical diagnostics. We ask:

Why does enlarging the tokenizer vocabulary improve Transformer performance, and which component of the loss benefits most?

Clarifying this mechanism is essential for principled tokenizer design and for understanding the true drivers of scaling laws in language modelling.

3 Experiments

Guiding Questions. Before diving into settings and metrics, we spell out the concrete questions that steer our empirical study:

1. **Tokenized Text Complexity**
   Does enlarging the vocabulary reduce the Kolmogorov complexity of tokenized text (3.3)?
2. **Skew vs. Segmentation**
   Does enlarging the vocabulary *mainly* sharpen the relative token-frequency distribution, or does it still increase single-token coverage of frequent words (§3.4)?
3. **Loss Decomposition**
   When the complexity decreases and frequency skew increases, how is cross-entropy reallocated between frequent and rare tokens, and which drives the global loss (§3.6)?
4. **Corpus Robustness**
   Are the above effects stable across different data quality—i.e. do high-curation (FineWeb-Edu) and noisier (OpenWebText) corpora exhibit the same trends (§3.5)?

The remainder of this section answers these questions in turn.

3.1 Experimental Settings

In these experiments, we train a byte-pair encoding (BPE) tokenizer [46] and estimate token frequencies using a sample of 10 billion GPT-2 tokens from FineWeb-Edu [37] and the entire OpenWebText [15]. For model pre-training, we use approximately 40 billion characters, about 7.5 billion tokens for FineWeb-Edu and 7 billion for OpenWebText with a 49K vocabulary. To compute the metrics below, we also drew an additional 5 billion characters that did not overlap with the training corpus. We report word-level average per-word loss to ensure fair comparison across vocabulary sizes. Whenever a smaller-vocabulary tokenizer splits a word into multiple tokens (i.e., subwords), we sum their individual losses. Our model comprises 85 million non-embedding parameters with pre-layer normalization (pre-LN) [52]. Training uses AdamW [29] ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$) with a learning rate of 6×10^{-4} that follows a cosine-decay schedule after a 350 million-token warmup, weight decay of 0.1, and gradient clipping at 1.0. Every experiment was repeated with five seeds.

3.2 Metrics

Kolmogorov Complexity of Tokenized Text Kolmogorov complexity $K(X)$ measures the minimal description length of a bitstring X and thus captures its inherent structure and compressibility [26, 16]. However, exact Kolmogorov complexity is uncomputable, since any such procedure would decide the halting problem [25, 33, 16]. Instead, we calculate a computable upper bound on Kolmogorov complexity, which serves as a practical metric for measuring the complexity of compressed data [16]. As BPE tokenization [46] is derived from a BPE compression algorithm [14], tokenized text can be viewed as compressed data. For a BPE-tokenized bitstring X^N, an asymptotic upper bound on $K(X^N)$ is given by

$$K(X^N) \;\le\; N\,H(p) \;+\; V\log_2 N \;+\; O(\log N), \tag{1}$$

where N is the total token count of tokenized text, $H(p)$ is the Shannon entropy of the token distribution, and the $V \log_2 N$ term accounts for the prefix-free encoding of the token (e.g., token frequency table). Since modern language models are trained on a massive corpus with billions of tokens, the $N\,H(p)$ component dominates, yielding the practical approximation $K(X^N) \approx N\,H(p)$.

Loss Decomposition Metrics In these experiments, we calculate three metrics to assess the language model's performance both for individual vocabulary types and overall: (1) Total Loss, (2) Average Per-Word Loss, and (3) Global Cross-Entropy Loss.

For each vocabulary v, we accumulate its *Total Loss*:

$$\text{Total Loss}(v) \;=\; \sum_{t \in N} \sum_{i=1}^{|t|} \mathbb{1}(v = t_i) \left[-\ln p(t_i \mid t_{<i}) \right], \tag{2}$$

where N is the set of evaluation documents, $\mathbb{1}(v = t_i)$ is the indicator that the ith token equals vocabulary v, and $-\ln p(t_i \mid t_{<i})$ is the negative log-likelihood of that token. Total loss measures the sum of negative log-likelihoods for each vocabulary, reflecting its impact on the model's loss [31].

Average Per-Word Loss for vocabulary v is defined as

$$\mu(\ell_v) \;=\; \frac{\text{Total Loss}(v)}{T_v(N)}, \tag{3}$$

where $T_v(N)$ is the total count of occurrences of v in the training data. This represents the mean negative log-likelihood across all occurrences of each vocabulary v [31].

The *Global Cross-Entropy Loss* is the weighted average of these per-word losses:

$$\text{Global Cross Entropy Loss} = \sum_{v} \frac{T_v(N)}{T(N)} \, \mu(\ell_v), \tag{4}$$

where $T(N) = \sum_v T_v(N)$ is the total token count in the training data. Global Cross-Entropy Loss reflects the model's average uncertainty per prediction [22, 31].

3.3 Increasing vocabulary size reduces complexity of tokenized text

Naturally occurring data contains a shared structure and inherently low complexity, models can learn effectively from training data, achieving high accuracy across diverse tasks and generalizing to unseen datasets [16]. To quantify how larger vocabularies make text more structured and compressible – better approximating natural data's low intrinsic complexity – we measure the upper bound of Kolmogorov complexity and the normalized compression ratio (NCR). NCR is a practical compressibility metric $\text{NCR}(x; C) \;=\; \frac{|C(x)|}{|x|}$ where $|x|$ is the byte length of data x and $|C(x)|$ is its size after lossless compression by compressor C; in our formulation, $K(X^N) \approx |C(x)| = N\,H(p)$. Table 1 reports the upper bound on Kolmogorov complexity and NCR for the 45.7 billion-byte FineWeb-Edu corpus. The results show that BPE tokenizers with larger vocabularies yield lower complexity and NCR, resulting in more structured and compressible data. This facilitates models to learn patterns in data more easily, as non-i.i.d. character sequences in text (e.g., words) are segmented as a single token, bringing a tokenized text sequence closer to i.i.d. and simplifying language modeling.

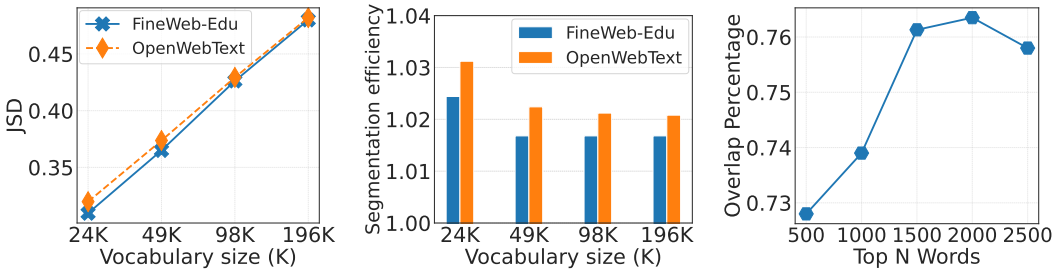
Table 1: Upper bound of Kolmogorov complexity ($K(X^N)$) and NCR for the 45.7 billion-byte FineWeb-Edu corpus with $K(X^N)$ measured in bytes.

Vocab size	$K(X^N)$	**NCR**
24K	$10.74B$	0.235
49K	$10.43B$	0.228
98K	$10.23B$	0.224
196K	$10.16B$	0.222

3.4 Segmentation already saturates; vocabulary growth mainly sharpens frequency skew

To quantify how each factor evolves with vocabulary growth, we disentangle two factors: (i) relative token-frequency imbalance (Figure 1a) and (ii) segmentation efficiency (Figure 1b)—the fraction of frequent words that are represented by exactly one token. *Frequent words* are operationally defined as the top $2,500$ word types, chosen because this cutoff already covers $\geq 70\%$ of corpus tokens in both FineWeb-Edu (74.4 %) and OpenWebText (75.5 %; full coverage curves in Appendix B).

In this experiment, relative token-frequency imbalance is quantified by the Jensen–Shannon divergence (JSD) from a uniform distribution of the same vocabulary size for a fair comparison across different vocabulary sizes [56, 12, 44]. Segmentation efficiency gauges the average token count per word, that is, how many tokens the tokenizer needs, on average, to encode each of the $2,500$ most frequent words. Figure 1a shows that JSD grows monotonically with vocabulary size, reflecting an increasingly heavy-tailed relative token-frequency distribution. In contrast, Figure 1b indicates that



(a) Token-frequency imbalance (b) Number of tokens per word (c) Overlap of frequent N words

Figure 1: Figure 1a shows that increasing vocabulary size exacerbates relative token-frequency imbalance. In other words, enlarging the vocabulary size introduces more rare tokens, causing the relative token-frequency distribution to be further from a uniform distribution. Figure 1b reveals that a 24K vocabulary size tokenizer already segments $2,500$ frequent words as a single token regardless of dataset quality. This implies that further vocabulary growth offers no added benefit for estimating the probabilities of frequent words. Figure 1c shows that the most frequent n words in fineWeb-Edu and OpenWebText largely overlap, highlighting the universality of frequent vocabulary across different datasets. We report the most frequent $2,500$ words in FineWeb-Edu and OpenWebText, which account for approximately 74.4% and 75.5% of each dataset, respectively.

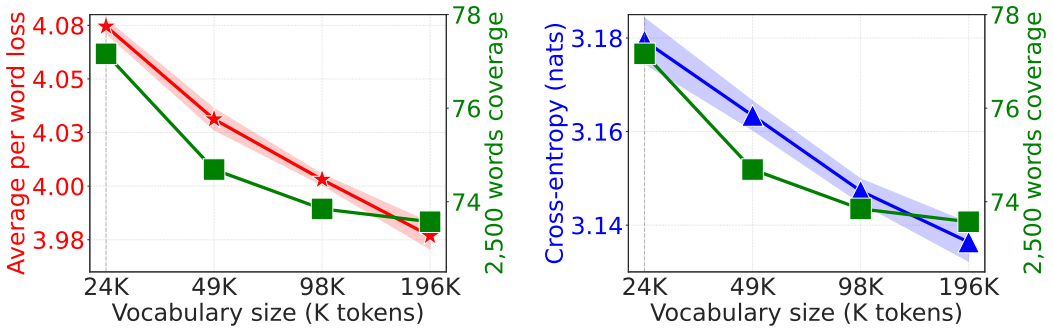
the segmentation efficiency exceeds 95 % by 24K tokens and plateaus later. In other words, beyond 24K, the vocabulary gets larger *without* providing additional single-token coverage for frequent words. Within the widely used $24\text{K}-196\text{K}$ range [43, 1, 41], enlarging the vocabulary chiefly amplifies relative token-frequency imbalance rather than improving segmentation efficiency. This finding tempers the common view that "bigger vocabularies help by approximating word-level tokens" [40]: that mechanism appears to saturate once frequent words are already tokenized as a single token.

3.5 Skew-driven effects persist across corpora quality levels

To test whether the effects of increasing vocabulary size hold across datasets of varying quality, we conduct experiments on both FineWeb-Edu [37] and OpenWebText [15]. Figure 1a and 1b demonstrate that the impact of increasing vocabulary size has a similar effect on both high-quality datasets (e.g., FineWeb-Edu [37]) and the lower-quality ones (e.g., OpenWebText [15]). Figure 1c further shows that the frequent $2,500$ words coincide with nearly 75% each other, highlighting the universality of frequent vocabulary across different corpora. Overall, the findings indicate that vocabulary size expansion produces the same effects irrespective of corpus quality and that high-frequency words substantially overlap across datasets.

3.6 Global loss declines as lower complexity and sharper skew cut frequent-token uncertainty

The effect of tokenized text complexity and relative token-frequency imbalance on language models has not yet been explored. Each prediction in a language model depends on previous tokens, and the conditional nature of these probabilities adds significant analytical complexity. However, it is possible to hypothesize that loss of rare token prediction weighs heavily in the overall loss, as conditional probability is often orders of magnitude lower and therefore yields much higher per-token losses [38, 32]. To address this presumption, we empirically examine how reducing loss on frequent word prediction at the expense of rare word predictions affects overall model performance. Figure 2a shows that enlarging the vocabulary steadily lowers the average per-word loss for the frequent words, where the gap between 24K and 196K is roughly 0.1 nats for the frequent $2,500$ words. Moreover, frequent words dominate the cross-entropy loss: frequent $2,500$ words contribute almost 80% of the total loss regardless of vocabulary size. However, as the vocabulary expands, the loss associated with infrequent words also rises, reflecting their diminished conditional probabilities and degraded predictability on those tokens. Despite this penalty, Figure 2b reports an overall decline in global cross-entropy loss from roughly 3.179 to 3.136 nats as the vocabulary size grows, indicating that the benefits of reducing loss on frequent words prediction outweigh the drawbacks of poorer infrequent word estimates. These observations highlight a key takeaway: lower tokenized text complexity makes the model predict frequent words with less uncertainty, and since the overall objective is governed by their loss contribution, further skewness in the relative token-frequency distribution generally reduces



(a) Frequent word loss on a 10B token FineWeb-Edu (b) Global cross-entropy on a 10B token FineWeb-Edu

Figure 2: Figure 2a illustrates that models with a larger vocabulary size reduce loss on the most frequent $2,500$ words in the FineWeb-Edu. It also reveals that frequent words account for nearly 80% of the total loss, while loss on infrequent words grows with vocabulary size as their conditional probabilities fall due to data sparsity. Nevertheless, Figure 2b shows that the global cross-entropy loss declines as vocabulary size increases, demonstrating that the gains from lower loss on frequent words outweigh the losses from poorer infrequent word estimates.

the data complexity and cross-entropy loss even though it increases rare words' losses. This finding is consistent with the existing study [40] on unigram models, explaining the benefits of increasing vocabulary size. We observe the same qualitative pattern in the OpenWebText (Appendix C).

4 Analysis

Although our results show that minimizing loss on high frequency tokens is crucial for reducing global cross-entropy, several questions remain unresolved. To pinpoint the causal chain from tokenizer design to model behavior, we now organize our analysis around three additional guiding questions.

Guiding Questions.

1. **Norm-Constraining and token-frequency Imbalance**
   If we constrain input/output embedding norms to eliminate the impact of token-frequency imbalance, does cross-entropy loss *increase* (§4.1)?
2. **Transfer Mechanism**
   How does the heavy overlap of the most frequent $2,500$ words connect pre-training loss drops to downstream accuracy (§4.2)?
3. **Parameter Count vs. Vocabulary Scaling**
   Can enlarging model size (with a fixed tokenizer) reproduce the same advantage delivered by a larger vocabulary (§4.3)?

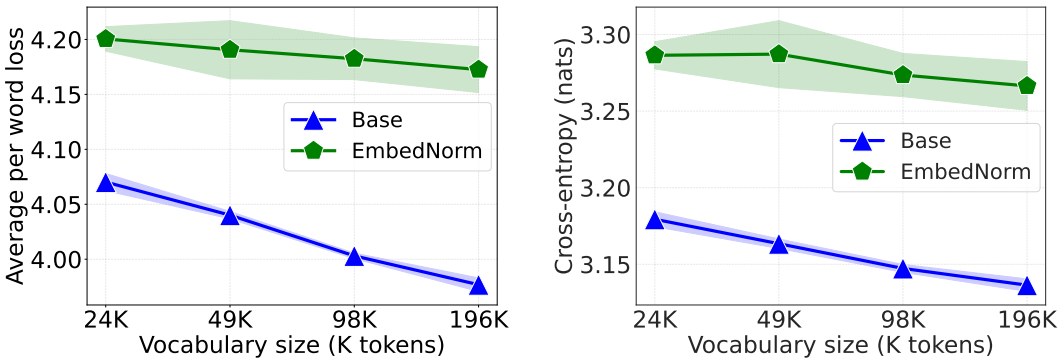
The next three subsections answer newly added Q1–Q3 in turn.

4.1 Constraining embedding norms *increases* cross-entropy

Higher token-frequency imbalance provides plentiful training examples for frequent words but far fewer for rare ones. Section 3.6 demonstrates that this reduces loss on frequent words while inflating it on rare words. We evaluate whether reducing the impact of token-frequency imbalance during pre-training leads to a performance degradation.

In language models, the token embedding norm of input and output embeddings reflects the token-frequency imbalance in training data. For input embeddings, every time a token shows up in a training batch, its token embedding receives gradient updates, and tokens that do not appear only undergo weight decay, causing their embedding norms to shrink over time [27, 10]. In the output embedding layer, the target token in the output embeddings receives a stronger gradient than non target token embedding during training [32] [3] (see the Appendix D for a detail explanation). Thus

[3] we assume input and output embeddings are untied from each other



(a) Frequent word loss gap of base and embednorm

(b) Global cross-entropy gap of base and embednorm

Figure 3: Constraining the norms of both input and output token embeddings to mitigate the impact of token-frequency imbalance during training increases average per-word loss on $2,500$ frequent words in the FineWeb-Edu as illustrated in figure 3a, which in turn drives up the global cross-entropy loss on the FineWeb-Edu (figure 3b).

embedding norm of a frequent token is typically larger in both input and output embedding. Even though language models with pre-LN [52] normalize the final layer hidden states, frequent tokens still develop larger output-embedding norms and highly align with the final layer hidden states, inflating their logits and predicted probabilities and ultimately reducing their average per-word loss.

To reduce the impact of token-frequency imbalance during pre-training, we constrain the token embeddings of the input and output embedding layer to a unit norm, since the token embeddings of our base model already average unit length. Figure 3a demonstrates that constraining the norms of both input and output token embeddings to address token-frequency imbalance during training leads to a higher average loss for $2,500$ frequent words. This increases the global cross-entropy on the training dataset and lowers downstream performance as illustrated in figure 3b and figure 4c respectively. This observation demonstrates that placing equal emphasis on rare and frequent tokens undermines performance; ensuring minimal uncertainty when predicting frequent tokens is essential.

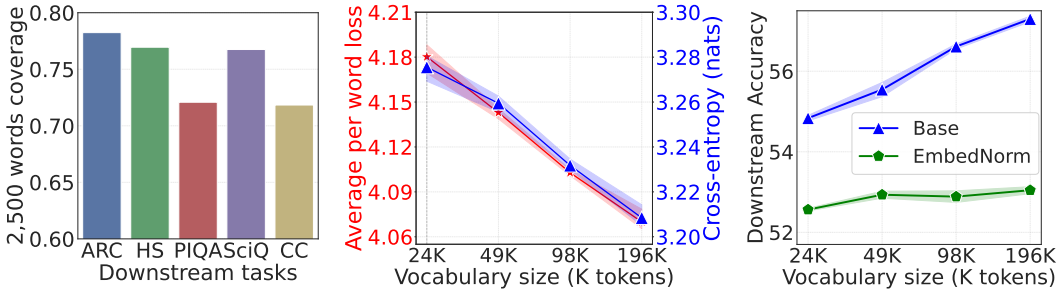
4.2 Frequent-word overlap explains downstream performance transfer

Reducing loss for frequent words is key to achieving a lower global cross-entropy loss. But does this effect carry over to downstream task accuracy? Although a strong link between lower loss and better downstream performance has been studied [20, 23], its rationale has not been investigated. To shed light on this phenomenon, we analyze the overlap of frequent words between the pre-training data and evaluation benchmarks and demonstrate that scaling the vocabulary size reduces cross-entropy loss during pre-training and on downstream tasks.

Figure 4a shows that the $2,500$ most common words in FineWeb-Edu [37] account for roughly 76-78% of all tokens in ARC [3], HellaSwag [55], and SciQ [51] and about 72% in the PIQA [6] and CC-Main-2023-40 dump [21]. Because lower cross-entropy loss on this CC subset correlates closely with stronger reasoning benchmark scores [21], reducing the frequent words loss and driving down global cross-entropy will improve accuracy across downstream tasks. Consistent with that expectation, figure 4b shows that increasing the vocabulary size from 24K to 196K lowers the average per-word loss on the frequent $2,500$ FineWeb-Edu words by about 0.11 nats, translating into a roughly 0.07 nats drop in global cross-entropy loss on the CC dataset. Figure 4c then confirms that this larger vocabulary also improves average downstream task accuracy of the language model from 54.5 to 57. The key insight from this observation is that frequent words highly overlap between the typical training dataset and the downstream benchmark, so the cross-entropy reduction achieved by enlarging the vocabulary size during pre-training naturally translates into better downstream performance.

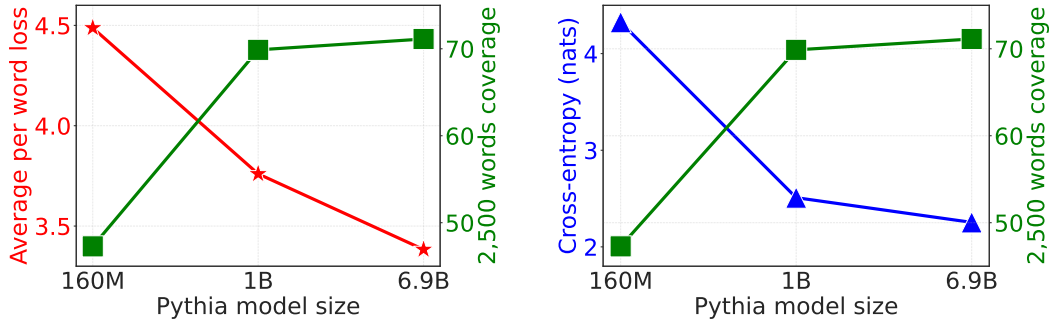
4.3 Parameter scaling recovers the same frequent-word gains

Thus far, we have shown that enlarging the vocabulary size decreases word-level average per-word loss on frequent words, which translates into lower cross-entropy loss and better downstream task



(a) Frequent word coverage (b) Frequent word loss reduction (c) Downstream performance

Figure 4: Figure 4a demonstrates that the most frequent $2,500$ words in the FineWeb-Edu comprise nearly $72-78\%$ of the tokens in other downstream benchmark datasets as well as the CC-Main-$2023-40$ [21]. ARC refers to ARC-Easy, and HS refers to HellaSwag. Figure 4b illustrates that a larger vocabulary reduces average per-word loss on frequent FineWeb-Edu words within the CC dataset, and demonstrates how this translates into lower global cross-entropy loss on CC dataset. Figure 4c confirms that scaling the vocabulary size boosts downstream task performance.



(a) Frequent word loss in pythia models (b) Global cross-entropy in pythia models

Figure 5: Figure 5a illustrates that increasing model size reduces loss on high frequency words, and the global cross-entropy loss of larger models is overwhelmingly driven by frequent word losses mirroring the effect of increased vocabulary size. However, unlike the pattern in figure 2a, scaling up model size does not exacerbate errors on infrequent tokens. Figure 5b demonstrates that the global cross-entropy loss declines as model size increases, showing the same tendency of scaling up the vocabulary size (figure 2b).

performance. One might ask whether similar gains could be achieved without altering the vocabulary size by adjusting other model hyperparameters. Our experiment illustrates that increasing the model's parameter count can replicate the same benefit. To investigate this, we use the Pythia suite, where each checkpoint is trained on the same dataset with identical hyperparameters, differing only in model size [4]. We also examine the OLMo-2 series [34] to determine whether the same pattern persists in contemporary large-scale language models (see the Appendix E).

Figure 5 compares Pythia models (160M, 1B, 6.9B parameter count) in terms of word-level average per-word loss, proportion of frequent words losses on total loss, and global cross-entropy loss measured on the Paloma validation dataset [31]. Figure 5a shows that larger models predict frequent words far more accurately: for the most frequent $2,500$ words in FineWeb-Edu, the average per-word loss drops from 4.48 nats (160M) to 3.76 nats (1B) and 3.38 nats (6.9B). Furthermore, frequent words loss dominates the total loss as model size grows, which accounts for almost 70% of the total loss in the Pythia 1B and 6.9B model size, compared with roughly 48% in the Pythia 160M. Unlike the trend observed when only the vocabulary size is increased (Figure 2b), scaling model capacity does not inflate loss on rare tokens; their share of the loss shrinks. Figure 5b illustrates that the global cross-entropy loss on the Paloma validation set drops from 4.32 nats in the Pythia 160M to 2.51 nats and 2.26 nats in the Pythia 1B and 6.9B respectively, mirroring the loss reduction seen when expanding the vocabulary (Figure 3b). From this experiment, we can observe that enlarging the

model size also lowers loss on frequent words while effectively ignoring rare token losses, so the global cross-entropy reduction is larger than what scaling up vocabulary size alone can deliver.

5 Discussion

5.1 Can we reduce the tokenized text complexity without intensifying frequency imbalance?

A larger vocabulary reduces the tokenized text complexity by decreasing the number of tokens and sharpening the token-frequency imbalance. However, section 3.6 shows that a sharper token-frequency imbalance increases rare-token loss. This raises the question: can we lower tokenized text complexity without enlarging the vocabulary size? SuperBPE [28] is one such example.

SuperBPE employs a two-stage BPE algorithm: in the first stage (up to a threshold vocabulary size t), it operates identically to standard BPE, while in the second stage, it abandons whitespace pre-tokenization. By permitting merges across former whitespace boundaries, Superbpe directs subsequent merges toward frequent tokens, thereby limiting the introduction of new rare tokens as the vocabulary expands, and preventing further decrease of relative token frequency of rare tokens. Simultaneously, it markedly decreases the token count of data, yielding up to 33% fewer tokens than standard BPE on average.

Table 2: Upper bound of Kolmogorov complexity ($K(X^N)$) and NCR for the 45.7 billion-byte FineWeb-Edu corpus tokenized with various SuperBPE variants. t and Avg denote the stage-switch vocabulary threshold and an average downstream performance reported by [28], respectively. The SuperBPE variant with the highest average performance exhibits the lowest tokenized text complexity.

superBPE	$K(X^N)$	**NCR**	**Avg [28]**
200K(BPE)	$10.21B$	0.223	39.8
200K($t = 180K$)	$10.03B$	0.219	43.8
200K($t = 160K$)	$10.05B$	0.220	43.4
200K($t = 80K$)	$10.10B$	0.221	42.9

Table 2 reports the upper bound Kolmogorov complexity, NCR of the 45.7 billion-byte FineWeb-Edu corpus tokenized with various SuperBPE variants, along with average downstream performance reported by [28]. The result shows that the superBPE achieves a lower token count and a less skewed token-frequency distribution than BPE at equal vocabulary. Also, the superBPE variant with the highest average performance exhibits the lowest tokenized text complexity. This finding demonstrates that both vocabulary expansion and the superBPE variants leverage the same benefit: they reduce the complexity of the tokenized text by segmenting common character sequences into single tokens, thereby facilitating the model to learn non-i.i.d. patterns in the data more easily.

5.2 Why higher token-frequency imbalance degrades machine translation task performance?

Zouhar et al. [56] reported that higher token-frequency imbalance degrades machine translation task performance, contrary to our findings. What explains this contrasting behavior between machine translation and monolingual settings? We presume that this is related to rare word issues in machine translation tasks [47, 45, 24, 30, 18].

In machine translation, the source and target vocabularies overlap minimally: vocabularies that are common in English often occur rarely in French, and vice versa. Conversely, in a monolingual setting, the pre-training corpus and downstream benchmarks share not only vocabularies but also many of the same frequent words, resulting in extensive text overlap. Consequently, BLEU penalizes every missing target-side n-gram, so mistranslating even a handful of infrequent tokens can sharply lower scores. To address this rare-word challenge, methods like vocabulary trimming deliberately shrink the vocabulary size to reduce the impact of low-frequency tokens on translation performance [17, 39, 9].

6 Related Work

Impact of tokenization on transformer language models Empirical and theoretical evidence shows that tokenization is a central determinant of both the quality and the speed with which Transformers learn. Early byte or character-level systems such as CANINE [11] and ByT5 [53] avoid sub-word vocabularies but pay a steep price: sequences become an order of magnitude longer, gradients are noisier, and convergence is markedly slower than for sub-word models. Enlarging the

BPE dictionary lets even an unigram model approach near-optimal cross-entropy by approximating word-level tokens, whereas the same model without a tokenizer underfits [40]. Furthermore, Controlled scaling studies reveal a log–linear pattern: exponentially expanding the input vocabulary leads to an almost linear drop in loss across model sizes [50]. Huang et al. [20] push this further with "Over-Tokenized Transformers," showing that a 400M parameter model with a 12.8M token encoder matches the loss of a 1B model baseline without extra compute.

Beyond the gains from larger vocabularies, a growing body of work shows that permitting merges across word boundaries (i.e., disabling whitespace pretokenization) improves language model performance. Liu et al. [28], Schmidt et al. [44] turn off white-space pretokenization during BPE tokenizer training once the vocabulary reaches t, or at frequency-driven transition points. Taken together, these results indicate that without tokenization, language models stall at character granularity, but with a tokenizer, they converge faster and generalize better.

Transformer language model as a context-aware lossless compressor Language modeling and lossless compression are effectively two views of the same task: modelling the next token distribution as sharply as possible so that an arithmetic or asymmetric-numeral coder can encode the symbol stream near its entropy rate [49, 13, 21]. This rationale came from Shannon's source–coding theorem [48], which states that the optimal average code-length for text sampled from a distribution p_{data} is $\mathbb{E}_{x\sim p_{\mathrm{data}}}\big[-\log_2 p_{\mathrm{data}}(x)\big]$, the smallest attainable number of bits for losslessly representing $x \sim p_{\mathrm{data}}$. When p_{data} is known, arithmetic coding [42, 36] achieves this bound by converting probabilities into bit-exact codes. In practice, however, p_{data} is unknown, so we substitute a language model $p_{\mathrm{model}}(x)$ as an approximation. Passing these model-derived probabilities to an arithmetic coder yields efficient compression. Consequently, lowering a language model's cross-entropy loss is equivalent to building a lossless compressor for the training corpus, with a Transformer acting as a context-sensitive probability estimator that drives the coding process.

7 Conclusion

This work set out to explain *how* and *why* larger vocabularies boost language model performance. Our experiments reveal a single, robust mechanism: enlarging the vocabulary reduces tokenized text complexity, making non-i.i.d. patterns easier to learn, better approximating natural data's low intrinsic complexity, and ultimately lowering language modeling difficulty. Once a vocabulary reaches roughly 24K size, every common word is already a single token; subsequent growth therefore does not refine segmentation but instead steepens the long-tailed frequency distribution, focusing optimization on the same frequent tokens and driving loss down. Norm-constraining that erases the frequency signal removes these gains, confirming causality, while enlarging model parameters with a fixed tokenizer reproduces the benefit, pointing to a shared optimization dynamic between vocabulary and model scaling. Recognising this fact turns Kolmogorov complexity of data into a principled dial for tokenizer–model co-design and sharpens our understanding of the scaling forces that govern language-model pre-training. We therefore conclude:

Expanding the tokenizer mainly reduces uncertainty for the most common words, with little payoff for the rare tail.

8 Limitations

Scaling up model size not only replicates the uncertainty reduction for frequent-word predictions achieved by a larger vocabulary, but also prevents the loss explosion on rare tokens, a pitfall of expanding the vocabulary size that ultimately yields diminishing returns in cross-entropy reduction. However, which scaling dimension – adding depth with extra layers, increasing width by enlarging hidden dimensions, or some other factor – drives this gain, and why, remains an open question. Answering these questions is crucial not only for understanding where the power of larger language models comes from but also for asking whether a carefully designed smaller language model trained from scratch can reproduce the performance of its much larger counterpart.

References

[1] M. Ali, M. Fromm, K. Thellmann, R. Rutmann, M. Lübbering, J. Leveling, K. Klug, J. Ebert, N. Doll, J. S. Buschhoff, C. Jain, A. A. Weber, L. Jurkschat, H. Abdelwahab, C. John, P. O. Suarez, M. Ostendorff, S. Weinbach, R. Sifa, S. Kesselheim, and N. Flores-Herr. Tokenizer choice for LLM training: Negligible or crucial? In K. Duh, H. Gómez-Adorno, and S. Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 3907–3924. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-NAACL.247. URL https://doi.org/10.18653/v1/2024.findings-naacl.247.

[2] Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/2000/hash/728f206c2a01bf572b5940d7d9a8fa4c-Abstract.html.

[3] S. Bhakthavatsalam, D. Khashabi, T. Khot, B. D. Mishra, K. Richardson, A. Sabharwal, C. Schoenick, O. Tafjord, and P. Clark. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge. *CoRR*, abs/2102.03315, 2021. URL https://arxiv.org/abs/2102.03315.

[4] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

[5] D. Bis, M. Podkorytov, and X. Liu. Too much in common: Shifting of embeddings in transformer language models and its implications. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5117–5130. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.403. URL https://doi.org/10.18653/v1/2021.naacl-main.403.

[6] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

[7] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, and S. Weinbach. Gpt-neox-20b: An open-source autoregressive language model. *CoRR*, abs/2204.06745, 2022. doi: 10.48550/ARXIV.2204.06745. URL https://doi.org/10.48550/arXiv.2204.06745.

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[9] P. Chizhov, C. Arnett, E. Korotkova, and I. P. Yamshchikov. BPE gets picky: Efficient vocabulary refinement during tokenizer training. In Y. Al-Onaizan, M. Bansal, and Y. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 16587–16604. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.925.

[10] W. Chung, J. Hong, N. M. An, J. Thorne, and S. Yun. Stable language model pre-training by reducing embedding variability. In Y. Al-Onaizan, M. Bansal, and Y. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 10852–10863. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.606.

[11] J. H. Clark, D. Garrette, I. Turc, and J. Wieting. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Trans. Assoc. Comput. Linguistics*, 10:73–91, 2022. doi: 10.1162/TACL_A_00448. URL https://doi.org/10.1162/tacl_a_00448.

[12] G. Dagan, G. Synnaeve, and B. Rozière. Getting the most out of your tokenizer for pre-training and domain adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ZFYBnLljtT.

[13] G. Delétang, A. Ruoss, P. Duquenne, E. Catt, T. Genewein, C. Mattern, J. Grau-Moya, L. K. Wenliang, M. Aitchison, L. Orseau, M. Hutter, and J. Veness. Language modeling is compression. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=jznbgiynus.

[14] P. Gage. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38, 1994. URL https://api.semanticscholar.org/CorpusID:59804030.

[15] A. Gokaslan, V. Cohen, E. Pavlick, and S. Tellex. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

[16] M. Goldblum, M. Finzi, K. Rowan, and A. G. Wilson. The no free lunch theorem, kolmogorov complexity, and the role of inductive biases in machine learning. *CoRR*, abs/2304.05366, 2023. doi: 10.48550/ARXIV.2304.05366. URL https://doi.org/10.48550/arXiv.2304.05366.

[17] T. Gowda and J. May. Finding the optimal vocabulary size for neural machine translation. In T. Cohn, Y. He, and Y. Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3955–3964. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.352. URL https://doi.org/10.18653/v1/2020.findings-emnlp.352.

[18] S. Gu, J. Zhang, F. Meng, Y. Feng, W. Xie, J. Zhou, and D. Yu. Token-level adaptive training for neural machine translation. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1035–1046. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.76. URL https://doi.org/10.18653/v1/2020.emnlp-main.76.

[19] J. Hayase, A. Liu, Y. Choi, S. Oh, and N. A. Smith. Data mixture inference attack: BPE tokenizers reveal training data compositions. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/10e6dfea9a673bef4a7b1cb9234891bc-Abstract-Conference.html.

[20] H. Huang, D. Zhu, B. Wu, Y. Zeng, Y. Wang, Q. Min, and X. Zhou. Over-tokenized transformer: Vocabulary is generally worth scaling. *CoRR*, abs/2501.16975, 2025. doi: 10.48550/ARXIV.2501.16975. URL https://doi.org/10.48550/arXiv.2501.16975.

[21] Y. Huang, J. Zhang, Z. Shan, and J. He. Compression represents intelligence linearly. *CoRR*, abs/2404.09937, 2024. doi: 10.48550/ARXIV.2404.09937. URL https://doi.org/10.48550/arXiv.2404.09937.

[22] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. M. Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *Journal of the Acoustical Society of America*, 62, 1977. URL https://api.semanticscholar.org/CorpusID:121680873.

[23] J. Kim, B. Lee, C. Park, Y. Oh, B. Kim, T. Yoo, S. Shin, D. Han, J. Shin, and K. M. Yoo. Peri-ln: Revisiting layer normalization in the transformer architecture. *CoRR*, abs/2502.02732, 2025. doi: 10.48550/ARXIV.2502.02732. URL https://doi.org/10.48550/arXiv.2502.02732.

[24] P. Koehn and R. Knowles. Six challenges for neural machine translation. In T. Luong, A. Birch, G. Neubig, and A. M. Finch, editors, *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics, 2017. doi: 10.18653/V1/W17-3204. URL https://doi.org/10.18653/v1/w17-3204.

[25] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics*, 2:157–168, 1968. URL https://api.semanticscholar.org/CorpusID:119745517.

[26] A. N. Kolmogorov. On tables of random numbers (reprinted from "sankhya: The indian journal of statistics", series a, vol. 25 part 4, 1963). *Theor. Comput. Sci.*, 207(2):387–395, 1998. doi: 10.1016/S0304-3975(98)00075-9. URL https://doi.org/10.1016/S0304-3975(98)00075-9.

[27] S. Land and M. Bartolo. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In Y. Al-Onaizan, M. Bansal, and Y. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 11631–11646. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.649.

[28] A. Liu, J. Hayase, V. Hofmann, S. Oh, N. A. Smith, and Y. Choi. Superbpe: Space travel for language models. *CoRR*, abs/2503.13423, 2025. doi: 10.48550/ARXIV.2503.13423. URL https://doi.org/10.48550/arXiv.2503.13423.

[29] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[30] T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 11–19. The Association for Computer Linguistics, 2015. doi: 10.3115/V1/P15-1002. URL https://doi.org/10.3115/v1/p15-1002.

[31] I. Magnusson, A. Bhagia, V. Hofmann, L. Soldaini, A. H. Jha, O. Tafjord, D. Schwenk, E. P. Walsh, Y. Elazar, K. Lo, D. Groeneveld, I. Beltagy, H. Hajishirzi, N. A. Smith, K. Richardson, and J. Dodge. Paloma: A benchmark for evaluating language model fit. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/760b2d94398aa61468aa3bc11506d9ea-Abstract-Datasets_and_Benchmarks_Track.html.

[32] A. Mircea, E. Lobacheva, and I. Rish. Gradient dissent in language model training and saturation. In *High-dimensional Learning Dynamics 2024: The Emergence of Structure and Reasoning*, 2024. URL https://openreview.net/forum?id=tJj3psv9nm.

[33] J. X. Morris, C. Sitawarin, C. Guo, N. Kokhlikyan, G. E. Suh, A. M. Rush, K. Chaudhuri, and S. Mahloujifar. How much do language models memorize? *CoRR*, abs/2505.24832, 2025. doi: 10.48550/ARXIV.2505.24832. URL https://doi.org/10.48550/arXiv.2505.24832.

[34] T. OLMo, P. Walsh, L. Soldaini, D. Groeneveld, K. Lo, S. Arora, A. Bhagia, Y. Gu, S. Huang, M. Jordan, N. Lambert, D. Schwenk, O. Tafjord, T. Anderson, D. Atkinson, F. Brahman, C. Clark, P. Dasigi, N. Dziri, M. Guerquin, H. Ivison, P. W. Koh, J. Liu, S. Malik, W. Merrill, L. J. V. Miranda, J. Morrison, T. Murray, C. Nam, V. Pyatkin, A. Rangapur, M. Schmitz, S. Skjonsberg, D. Wadden, C. Wilhelm, M. Wilson, L. Zettlemoyer, A. Farhadi, N. A. Smith, and H. Hajishirzi. 2 olmo 2 furious. *CoRR*, abs/2501.00656, 2025. doi: 10.48550/ARXIV.2501.00656. URL https://doi.org/10.48550/arXiv.2501.00656.

[35] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

[36] R. Pasco. Efficient coding of markov sources by arithmetic coding. In *Proceedings of the 1977 International Conference on Communications*, pages 51–55, 1977.

[37] G. Penedo, H. Kydlícek, L. B. Allal, A. Lozhkov, M. Mitchell, C. A. Raffel, L. von Werra, and T. Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

[38] A. Pinto, T. Galanti, and R. Balestriero. The fair language model paradox. *CoRR*, abs/2410.11985, 2024. doi: 10.48550/ARXIV.2410.11985. URL https://doi.org/10.48550/arXiv.2410.11985.

[39] I. Provilkov, D. Emelianenko, and E. Voita. Bpe-dropout: Simple and effective subword regularization. In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1882–1892. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.170. URL https://doi.org/10.18653/v1/2020.acl-main.170.

[40] N. Rajaraman, J. Jiao, and K. Ramchandran. An analysis of tokenization: Transformers under markov data. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/724afcaae4ae92a9220a077ffe80088d-Abstract-Conference.html.

[41] V. Reddy, C. W. Schmidt, Y. Pinter, and C. Tanner. How much is enough? the diminishing returns of tokenization training data. *CoRR*, abs/2502.20273, 2025. doi: 10.48550/ARXIV.2502.20273. URL https://doi.org/10.48550/arXiv.2502.20273.

[42] J. Rissanen. Generalized kraft inequality and arithmetic coding. *IBM J. Res. Dev.*, 20(3):198–203, 1976. doi: 10.1147/RD.203.0198. URL https://doi.org/10.1147/rd.203.0198.

[43] C. W. Schmidt, V. Reddy, H. Zhang, A. Alameddine, O. Uzan, Y. Pinter, and C. Tanner. Tokenization is more than compression. In Y. Al-Onaizan, M. Bansal, and Y. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 678–702. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.40.

[44] C. W. Schmidt, V. Reddy, C. Tanner, and Y. Pinter. Boundless byte pair encoding: Breaking the pre-tokenization barrier, 2025. URL https://arxiv.org/abs/2504.00178.

[45] R. Sennrich and B. Zhang. Revisiting low-resource neural machine translation: A case study. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 211–221. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1021. URL https://doi.org/10.18653/v1/p19-1021.

[46] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1162. URL https://doi.org/10.18653/v1/p16-1162.

[47] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1162. URL https://doi.org/10.18653/v1/p16-1162.

[48] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948. doi: 10.1002/J.1538-7305.1948.TB01338.X. URL https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[49] C. E. Shannon. Prediction and entropy of printed english. *The Bell System Technical Journal*, 30(1):50–64, 1951. doi: 10.1002/j.1538-7305.1951.tb01366.x.

[50] C. Tao, Q. Liu, L. Dou, N. Muennighoff, Z. Wan, P. Luo, M. Lin, and N. Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/cf5a019ae9c11b4be88213ce3f85d85c-Abstract-Conference.html.

[51] J. Welbl, N. F. Liu, and M. Gardner. Crowdsourcing multiple choice science questions. In L. Derczynski, W. Xu, A. Ritter, and T. Baldwin, editors, *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 94–106. Association for Computational Linguistics, 2017. doi: 10.18653/V1/W17-4413. URL https://doi.org/10.18653/v1/w17-4413.

[52] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR, 2020. URL http://proceedings.mlr.press/v119/xiong20b.html.

[53] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Trans. Assoc. Comput. Linguistics*, 10:291–306, 2022. doi: 10.1162/TACL_A_00461. URL https://doi.org/10.1162/tacl_a_00461.

[54] D. Yu, E. Cohen, B. Ghazi, Y. Huang, P. Kamath, R. Kumar, D. Liu, and C. Zhang. Scaling embedding layers in language models. *CoRR*, abs/2502.01637, 2025. doi: 10.48550/ARXIV.2502.01637. URL https://doi.org/10.48550/arXiv.2502.01637.

[55] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

[56] V. Zouhar, C. Meister, J. L. Gastaldi, L. Du, M. Sachan, and R. Cotterell. Tokenization and the noiseless channel. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5184–5207. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.284. URL https://doi.org/10.18653/v1/2023.acl-long.284.

A Influence of token-frequency imbalance on unigram and language models

In this section, we provide a detailed explanation of our research question. Natural language exhibits strong contextual dependencies rather than behaving as an i.i.d. process: each token's probability depends on its preceding context. As a result, the entropy rate, $H_\infty = \lim_{t\to\infty} H\big(X_t \mid X_{<t}\big)$, which captures the optimal per-token uncertainty in text, is strictly lower than the unigram Shannon entropy $H_1 = -\sum_w p(w)\,\log p(w)$. Although entropy rate and Shannon entropy coincide for truly i.i.d. data, in natural language, they can differ by several bits per token.

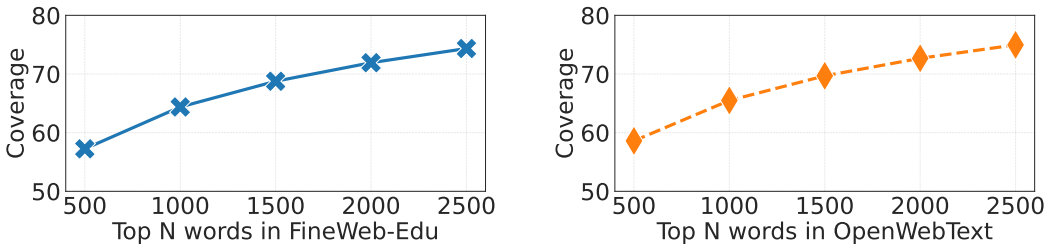
Under a pure unigram model, minimum cross-entropy loss exactly equals Shannon entropy, and modifying the vocabulary size of the tokenizer immediately changes the Shannon entropy. Typically, enlarging the vocabulary segments frequent multi-token patterns into single tokens, driving their individual relative frequency up and reducing Shannon entropy. But if the vocabulary size grows too large, the new entries tend to be rare tokens, which lengthen the tail and can actually increase the Shannon entropy as token-frequency imbalance rises [56]. However, experimental results show that expanding a BPE vocabulary to around 80k lowers the Shannon entropy of unigram models, demonstrating that a more skewed token-frequency distribution is advantageous at practical vocabulary scales [40]. This pattern extends to n-gram models as well: their conditional Shannon entropy—the lowest possible n-gram cross-entropy—can never exceed the unigram entropy, so lowering the unigram entropy necessarily lowers the conditional entropy.

Language model loss $\mathcal{L}(\theta)$ minimizes

$$\mathcal{L}(\theta) = H_\infty \;+\; \sum_{x\in V} p(x)\, D_{\mathrm{KL}}\big(P(\cdot \mid x_{<t}) \,\|\, Q_\theta(\cdot \mid x_{<t})\big). \tag{5}$$

where V denotes the tokenizer's vocabulary, $p(x)$ the marginal probability of token x, $P(\,\cdot \mid x_{<t})$ the true next-token distribution given the full history $x_{<t}$, and $Q_\theta(\,\cdot \mid x_{<t})$ the model's predicted distribution with parameters θ. When the target label is a one-hot vector, the language model loss can be written as $\mathcal{L}(\theta) = \sum_{x\in V} p(x)\left[-\log Q_\theta(x \mid x_{<t})\right]$ so it is a marginal-frequency-weighted average of the model's surprisal $-\log Q_\theta$. By contrast, the Shannon entropy of the unigram model is a marginal-frequency-weighted average of the self-information $-\log p(x)$ in the dataset. Even though frequent token logits and embedding norms are higher than rare ones (see §4.1 and Appendix D), the loss $-\log Q_\theta$ depends not only on the underlying relative token-frequencies but also on training dynamics as well. We therefore cannot derive a closed-form expression to predict how much the loss on frequent tokens shrinks versus how much the rare token losses grow under $\mathcal{L} = p_{\mathrm{freq}}\, L_{\mathrm{freq}} + p_{\mathrm{rare}}\, L_{\mathrm{rare}}$. This masking effect makes it substantially harder to measure the influence of token-frequency imbalance in language models than in the unigram and n-gram models.

B Coverage of the most frequent N words



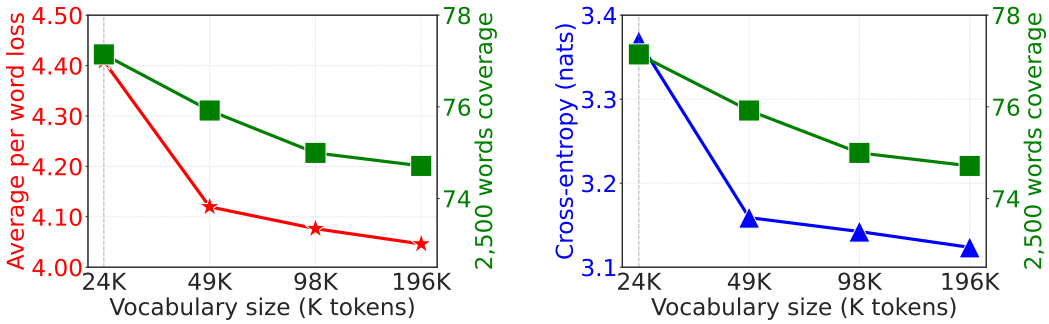
(a) Coverage of frequent words in FineWeb-Edu

(b) Coverage of frequent words in OpenWebText

Figure 6: Figures 6a and 6b illustrate the cumulative coverage of the 2,500 most frequent words in the Fineweb-edu and OpenWebText datasets, respectively.

In this section, we measure the coverage of the frequent words in Fineweb-edu [37] and OpenWebText [15]. Both dataset exhibit a steep rise in cumulative coverage as we include more high-frequency tokens, but with subtly different baselines and slopes. In figure 6a, the most frequent 500 words already cover about 58% of all tokens, climbing to roughly 75% once we take the $2,500$ most frequent words. OpenWebText (Figure 6b) starts marginally higher—around 59% at most frequent 500 words, but follows an almost identical trajectory, reaching about 76% coverage by the frequent $2,500$ words. This pattern underscores how a relatively small core vocabulary captures the vast majority of running text in both corpora, with only modest gains as we move deeper into the long tail.

C OpenWebText experiments results



(a) Frequent word loss on an OpenWebText

(b) Global cross-entropy on an OpenWebText

Figure 7: Figure 7a reveals that expanding the vocabulary from 24K to 196K steadily reduces the average per-vocabulary loss of high frequency words. Figure 7b indicates that the most frequent $2,500$ tokens still contribute roughly 75% of the total loss, while the rare words losses grow with vocabulary size, similar to 2b. Figure 7b further shows that the global cross-entropy loss falls by about 0.25 nats as the vocabulary grows, demonstrating that the reduction of loss on frequent words outweighs the inflation of rare-token losses.

To verify that reducing frequent-word loss is not a by-product of dataset quality, we repeat the same experiments in section 3.6 on the OpenWebText dataset. Figure 7a shows that widening the vocabulary from 24K to 196K in OpenWebText progressively reduces the average loss assigned to high-frequency words. Figure 7b indicates that the most frequent $2,500$ tokens still account for about 75% of the total loss, whereas the loss on infrequent tokens grows with vocabulary size, paralleling the pattern seen in Figure 2b. Figure 7b further demonstrates that the global cross-entropy loss falls from 3.37 nats at a 24K vocabulary to 3.12 nats at 196K, implying that the reduction in loss on frequent words more than offsets the increase in rare-token loss, regardless of dataset quality or type.

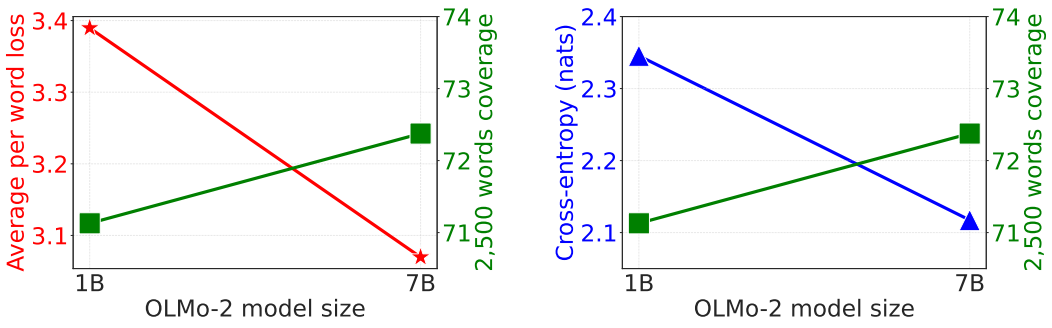
D Frequent and rare token norm in output embedding

In this section, we explain why frequent tokens acquire larger output-embedding norms and logits by deriving and analysing the gradients of the output embedding. Cross-entropy loss with a vocabulary size $|V|$ and hidden dimension size d_{model}, using a one-hot target $t \in \mathbb{R}^{|V|}$, the logit vector is $\ell = hE_{\text{out}}^{\top}$ where $h \in \mathbb{R}^{d_{\text{model}}}$ is the final hidden state and $E_{\text{out}} \in \mathbb{R}^{|V| \times d_{\text{model}}} = [u_1, \ldots, u_{|V|}]$ is the output-embedding matrix. When input and output embeddings are untied, the gradient with respect to each row of E_{out} decomposes into

$$\frac{\partial \mathcal{L}}{\partial E_{\text{out}_t}} = (p_t - 1)\, h, \qquad \frac{\partial \mathcal{L}}{\partial E_{\text{out}_j}} = p_j\, h \;\; (j \neq t), \tag{6}$$

where $p_t = \text{softmax}(\ell)_t$ [5, 32]. Because $p_t \ll 1$ at the start of training, $\left\|\partial \mathcal{L}/\partial E_{\text{out}_t}\right\|_2 \approx \|h\|_2$ while each non-target row scales only with $p_j \|h\|_2$ ($p_j < 1/|V|$). Thus, every time token t appears, its embedding is pulled almost $\|h\|_2$ units along $+h$, whereas each competing row is nudged by a factor of $p_j \ll 1$. As tokens recur in the training data, their token embeddings accumulate gradient updates roughly proportional to their counts, so $\|E_{\text{out}_t}\|_2$ grows in line with token-frequency. Since each logit factorizes as $\ell_t = \|h\|_2 \, \|E_{\text{out}_t}\|_2 \cos\theta_t$ and frequent tokens not only acquire the largest norms but also align closely with final hidden-state directions $\cos\theta_t \approx 1$, they end up with disproportionately large logits whereas rare tokens suffer both smaller norms and larger angular deviations [5, 27]. Although ℓ_2 weight decay can slow this norm inflation, they merely dampen the effect rather than remove the underlying frequency norm logit correlation. This frequency-proportional amplification explains the empirical observation that output-embedding norms and softmax logits scale with token-frequency in standard language models.

E OLMo-2 result



(a) Frequent word loss in OLMo-2 models

(b) Global cross-entropy in OLMo-2 models

Figure 8: Figure 8a illustrates that larger 7B model reduces average per-vocabulary loss from 3.39 nats to 3.07 nats while slightly increasing the proportion of loss covered by frequent words from 71% to 72.5%. Figure 8b further demonstrates that Scaling from 1B to 7B reduces the overall cross-entropy from 2.35 nats to 2.12 nats, confirming the same pattern persists in contemporary large-scale language models.

To identify whether the reduction in frequent words loss with increasing model size holds for contemporary large language models, we perform analogous experiments using the OLMo-2 series [34]. Figure 8a and 8b indicate that the average per vocabulary loss falls from 3.39 nats in the 1B parameter model to 3.07 nats in the 7B variant while slightly increasing the proportion of loss covered by $2,500$ frequent words from 71% to 72.5%. Figure 8b further shows that global cross-entropy loss declines from 2.35 nats for OLMo-2 1B to 2.12 nats for OLMo-2 7B. Notably, OLMo-2 employs a much larger vocabulary (cl100K [35]) than Pythia (50304 tokens [7]) and trains on a larger corpus [34], which helps drive down the average loss on high-frequency words. These results confirm that the same trend holds for modern large-scale language models as well.