Upsample or *Upweight?*
Balanced Training on Heavily Imbalanced Datasets

Tianjian Li, Haoran Xu, Weiting Tan
Kenton Murray, Daniel Khashabi
Center for Language and Speech Processing
Johns Hopkins University
tli104@jhu.edu

Abstract

Data abundance across different domains exhibits a long-tailed distribution: few domains have abundant data, while most face data scarcity. Our work focuses on a multilingual setting, where available data is heavily skewed towards high-resource languages. Two common strategies to address this disparity are upsampling low-resource data (*Temperature Sampling*) and upweighting low-resource loss (*Scalarization*). These methods are often assumed to be equivalent, but this equivalence has not been rigorously established, prompting our investigation.

Through theoretical and empirical analysis, we identify when these two methods are equivalent and when they diverge. We prove that they are equivalent under *full* gradient descent but differ under *stochastic* gradient descent due to differences in gradient variance. Specifically, *Temperature Sampling* exhibits lower variance in gradient estimation compared to *Scalarization*, leading to faster convergence but a higher risk of overfitting. Based on these insights, we propose COOLDOWN, a strategy that starts by heavily upsampling low-resource languages to accelerate convergence and gradually reduces the upsampling to prevent overfitting—achieving the best of both worlds. Our method competes effectively with existing data re-weighting techniques while offering computational efficiency.

1 Introduction

Information on the internet ranges from common knowledge, such as famous landmarks, to rare details, such as local folklore and specialized scientific theories. Data availability across different domains is often long-tailed (Feldman, 2020; Feldman and Zhang, 2020; Kandpal et al., 2023), where very few domains have abundant data. However, the standard language model training objective treats each training instance equally, putting no emphasis on domains that suffer from data scarcity.
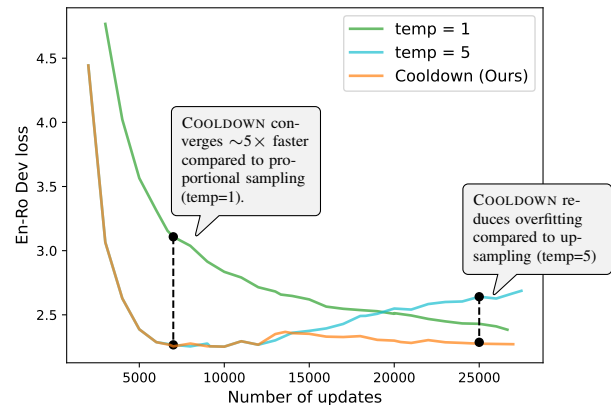

Figure 1: Validation loss by training iteration of a low-resource language pair (En-Ro) in multilingual machine translation. Proportional sampling leads to underfitting the low-resource direction. Using a high temperature (oversampling LRLs) leads to overfitting the low-resource direction. Employing a high temperature at the beginning and then decreasing the temperature (COOLDOWN) gets the advantage of fast convergence without overfitting.

This heavy mismatch in dataset sizes creates substantial challenges in training language models to be competent in all domains.

This work focuses on language modeling on a natural divide of domains with a heavy mismatch: different languages in multilingual language modeling. Multilingual language models are often trained on corpora with an overwhelming amount of English and other high-resource languages (HRLs) and tiny amounts of data for low-resource languages (LRLs) (Koehn and Knowles, 2017; Conneau et al., 2020; Xue et al., 2021). For example, the multilingual C4 corpus (Xue et al., 2021) contains 2733 billion English tokens but only 1 billion Swahili tokens. Uniformly sampling from the combined dataset would result in the language model optimized heavily towards performance on HRLs (e.g. English), sacrificing performance on LRLs

(e.g. Swahili).

Two methods are often employed to address domain mismatches: **Scalarization** and **Temperature Sampling**. Scalarization adjusts the losses for individual domains by re-weighting them under uniform sampling (Zhou et al., 2021; Choi et al., 2023). In this case, we assign a larger weight to LRLs to emphasize their importance. Temperature Sampling, weights each training instance uniformly and handles the mismatch by over-sampling LRLs and/or down-sampling HRLs (Aharoni et al., 2019; Wang et al., 2020b; Chung et al., 2023; Xue et al., 2021). Intuitively, Scalarization modifies the loss while Temperature Sampling modifies the dataset. Scalarization and Temperature Sampling are widely regarded as equivalent. Choi et al. (2023) denotes *"we follow convention and implement Scalarization via proportional sampling"*. Xie et al. (2023) and Fan et al. (2024) implement sampling probabilities by multiplying losses with per-domain re-normalized weights. The underlying assumption is that Temperature Sampling and Scalarization are equivalent, and we can use them interchangeably. However, to the best of our knowledge, this equivalence has not been rigorously established.

We closely investigate this assumed equivalency in theory (§3.1). Specifically, we prove that although they are equivalent in *full* gradient descent (Theorem 1), Temperature Sampling induces lower variance in the context of *stochastic* gradient descent (Theorem 2). Moreover, the variance induced by scalarization increases as the approximated temperature increases or the domain distribution's skewness increases (Theorem 3). Based on our theoretical results and connecting to the literature on lower variance between stochastic gradients accelerates convergence (Sutskever et al., 2013; McCandlish et al., 2018), we make the following hypothesis:

> **Hypothesis 1.** Temperature Sampling converges much faster than Scalarization at higher temperatures or on heavily imbalanced domain distributions.

We empirically verify our hypothesis (§3.2) and find that Temperature Sampling does converge faster but is more prone to overfitting. We identify that the temperature controls the speed of convergence and hence can be used as a control knob to adjust the convergence speed. We thus propose COOLDOWN: to use a large temperature initially for fast convergence, then decrease the temperature to prevent overfitting to the LRLs. Figure 1 illustrates the effectiveness of COOLDOWN, which significantly accelerates convergence on the LRL due to a high temperature (aggressive upsampling of LRLs) at the beginning of training and reduces overfitting to the LRL due to the lowering the temperature during training.

To sum up, our contribution is two-fold:

- We inspect Scalarization and Temperature Sampling both theoretically and empirically (§3). Contrary to existing work that uses them interchangeably, we found that Temperature Sampling converges faster due to a lower variance in stochastic gradient estimation.

- Motivated by our findings, we propose COOLDOWN, a method to adjust the sampling temperature during training on unbalanced datasets. We show the effectiveness of COOLDOWN in multilingual settings.

2 Preliminaries

2.1 Notations and Task Description

We consider a model trained on a collection of data $\mathcal{D} = \{x\}_{i=1}^N$ from K domains $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup ... \cup \mathcal{D}_K$. Here, "domain" refers to sources (Books, Wikipedia, code) for general language modeling or different languages (English, French, Swahili) in multilingual language modeling. The total training loss $\mathcal{J}(\mathcal{D})$ is the sum of the losses of each example $\mathcal{L}(x)$.

$$\mathcal{J}(\mathcal{D}) = \sum_{x \in \mathcal{D}} \mathcal{L}(x).$$

Scalarization (S) Naive aggregation often results in imbalanced performance across domains when high-resource domains dominate the aggregated loss. Scalarization solves this issue by assigning weights $\mathbf{w} = \{w_i\}_{i=1}^K$ to each domain and aggregates the weighted sum of individual losses:

$$\mathcal{L}_S(\mathbf{w}) = \mathbb{E}_{x \in \mathcal{D}} \left[w_{f(x)} \mathcal{L}(x) \right],$$

where $f : \mathcal{D} \to [K]$ maps a training example to the index of its domain. Scalarization balances the loss by assigning a higher weight to harder or low-resource domains.

Temperature Sampling (TS) Instead of assigning weights to losses, we can also sample more frequently from the low-resource domain to achieve balanced training. Temperature sampling achieves

this by adjusting the probabilities of selecting instances from different domains based on their sizes. The sampling probability vector p of each domain is given by:

$$\forall i \in \{1, 2, ..., K\} : \ p(i; \tau) = \frac{|\mathcal{D}_i|^{\frac{1}{\tau}}}{\sum_{j=1}^{K} |\mathcal{D}_j|^{\frac{1}{\tau}}},$$

where τ is the sampling temperature, a hyperparameter controlling the sampling weights. $\tau = 1$ means that we are sampling proportional to the sizes of each domain. As we increase τ, we increase the sampling probability of low-resource domains. The loss for Temperature Sampling is:

$$\mathcal{L}_{TS}(\tau) = \mathop{\mathbb{E}}_{\substack{k \sim p(\cdot; \tau) \\ x \sim \mathcal{D}_k}} \left[\mathcal{L}(x) \right]$$

The common understanding is that Temperature Sampling is mathematically equivalent to Scalarization (Choi et al., 2023; Xin et al., 2022), and we can use them interchangeably (Choi et al., 2023). In the next subsection, we will formalize this statement and show that these two are mathematically equivalent in *full* gradient descent. We will then show that they are <u>not</u> equivalent under *stochastic* gradient descent.

3 Temperature Sampling v.s. Scalarization

3.1 Theoretical Analysis

We formalize the equivalence of Scalarization and weighted sampling under full-gradient descent (Theorem 1) and show that Scalarization induces a larger variance between the mini-batch losses (Theorem 2). Furthermore, when using Scalarization to approximate Temperature Sampling, the variance increases as the temperature rises (Theorem 3).

Theorem 1 (Equivalency under Gradient Descent).
For any sampling temperature τ, there exists a set of weights $\mathbf{w}_\tau = \{w_1, w_2, ..., w_K\}$ for the Scalarization loss such that this loss is equivalent to the Temperature Sampling loss, both computed based on the whole data $\mathcal{D}$.

Proof. For all $i \in \{1, 2, 3, ..., K\}$, let:

$$w_i = \frac{\sum_{j=1}^{K} |\mathcal{D}_j|}{|\mathcal{D}_i|} \cdot \frac{|\mathcal{D}_i|^{\frac{1}{\tau}}}{\sum_{j=1}^{K} |\mathcal{D}_j|^{\frac{1}{\tau}}} = \frac{p(i; \tau)}{p(i; 1)},$$

$$\begin{aligned}
\mathcal{L}_S(\mathbf{w}_\tau) &= \mathbb{E}_{x \sim \mathcal{D}} \left[w_{f(x)} \mathcal{L}(x) \right] \\
&= \sum_{x \in \mathcal{D}} \frac{p(f(x); 1)}{|\mathcal{D}_{f(x)}|} w_{f(x)} \mathcal{L}(x) \\
&= \sum_{i=1}^{K} p(i; \tau) \sum_{x \in \mathcal{D}_i} \frac{1}{|\mathcal{D}_i|} \mathcal{L}(x) \\
&= \mathop{\mathbb{E}}_{i \sim p(\cdot; \tau)} \left[\sum_{x \in \mathcal{D}_i} \frac{1}{|\mathcal{D}_i|} \mathcal{L}(x) \right] \\
&= \mathbb{E}_{i \sim p(\cdot; \tau)} [\mathbb{E}_{x \sim \mathcal{D}_i} [\mathcal{L}(x)]] = \mathcal{L}_{TS}(\tau).
\end{aligned}$$

$\square$

Intuitively, this suggests that in the context of *full* gradient descent, the loss remains the same whether you multiply the loss of a single data point by 2 (S) or duplicate the data point (TS). We will then show that Scalarization induces a larger variance in *stochastic* gradient estimation (Robbins, 1951) compared to Temperature Sampling.

Corollary 1.1. *For any τ, let $\mathbf{w}_\tau$ be the set of weights such that $\mathcal{L}_{TS}(\tau) = \mathcal{L}_S(\mathbf{w}_\tau)$. Let $\nabla \mathcal{L}(x)$ be the gradient with respect to a single datapoint x. We denote the stochastic gradient under Scalarization as $\nabla \mathcal{L}_S(x; \mathbf{w}_\tau) = \{\nabla w_{f(x)} \mathcal{L}(x) | x \sim \mathcal{D}\}$, which is the gradient $\nabla w_{f(x)} \mathcal{L}(x)$ when a single sample x is uniformly drawn from the dataset $\mathcal{D}$. Similarly, we denote the stochastic gradient under Temperature Sampling as $\nabla \mathcal{L}_{TS}(x; \tau) = \{\nabla \mathcal{L}(x) | x \sim \mathcal{D}_i, i \sim p(i; \tau)\}$. Then both $\nabla \mathcal{L}_S(x; \mathbf{w}_\tau)$ and $\nabla \mathcal{L}_{TS}(x; \tau)$ are unbiased estimates of the total gradient.*

Proof. By definition, we have

$$\begin{aligned}
\mathbb{E}[\nabla \mathcal{L}_s(x; \mathbf{w}_\tau)] &= \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[w_{f(x)} \nabla \mathcal{L}(x) \right] \\
&= \mathbb{E}_{\mathcal{D}_i \sim p(\cdot; \tau)} [\mathbb{E}_{x \sim \mathcal{D}_i} [\nabla \mathcal{L}(x)]] \\
&= \mathbb{E}[\nabla \mathcal{L}_{TS}(x; \tau)]
\end{aligned}$$

$\square$

Theorem 2 (Scalarization induces larger variance under Stochastic Gradient Descent). *Using the same notation in Corollary 1.1, we have $\text{Var}(\nabla \mathcal{L}_S(x; \mathbf{w}_\tau)) \geq \text{Var}(\nabla \mathcal{L}_{TS}(x; \tau))$.*

We defer the proof of Theorem 2 to Appendix B.

Implication of Theorem 2 Scalarization induces larger variance connects with the literature on variance reduction in stochastic gradient estimation

(Sutskever et al., 2013; Kingma and Ba, 2015) accelerates the convergence of SGD (Robbins, 1951). We thus hypothesize that Temperature Sampling will converge faster than Scalarization with the set of weights that makes them mathematically equivalent under *full* gradient descent.

Theorem 3 (Scalarization induces larger variance when approximating higher temperatures). *The difference:*

$$\Delta = \text{Var}(\nabla \mathcal{L}_S(x; \mathbf{w}_\tau)) - \text{Var}(\nabla \mathcal{L}_{TS}(x; \tau))$$

monotonically increases when $\tau \geq 1$.

Figure 2 illustrates the variance by increasing temperature and skewness of the distribution. We defer details on how we constructed Figure 2 and the full proof of Theorem 3 to Appendix C.
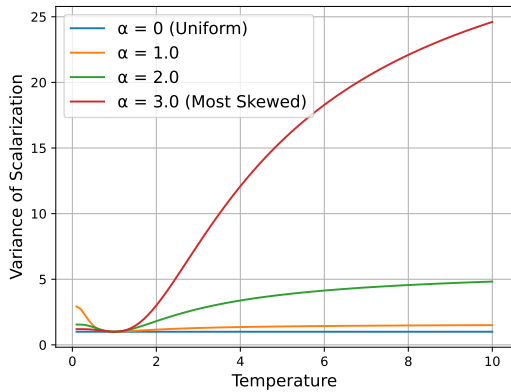

Figure 2: Variance of Scalarization $\sum_i \frac{p(i;\tau)^2}{p(i;1)}$ by sampling temperature τ. A large temperature or a skewed distribution of $\mathcal{D}$ induces a much larger variance for Scalarization. Distributions $\mathcal{D}_i \propto \frac{1}{i^\alpha}$. See Appendix C for details of the experiment setup.

Implication of Theorem 3   The fact that the induced variance of Scalarization increases as the approximated temperature increases implies that Temperature Sampling converges much faster than Scalarization at higher temperatures, which we empirically verify in the next section.

3.2   Empirical Evidence

We directly validate our hypothesis that lower variance of Temperature Sampling accelerates convergence.[1] We train a multilingual machine translation model and vary the sampling temperature $\tau = \{2, 3, 5\}$. We then approximate Temperature Sampling by multiplying the Temperature Sampling probabilities by loss under proportional sam-

pling ($\tau = 1$). Specifically, we pair one high-resource direction in En-{Fr, Cs} with the low-resource direction En-Ro. We report the statistics of the datasets we used in §3.2 at Table 1, for all of our experiments, we learn a shared Byte-Pair Encoding tokenizer with a vocabulary size of 32k.[2]

Language Pairs	Training	Validation
WMT 15 En-Fr	40M	4503
WMT 22 En-Zh	55M	3418
WMT 22 En-Cs	56M	2082
WMT 16 En-Ro	600K	1999

Table 1: Dataset statistics for comparing Scalarization v.s. Temperature Sampling in §3.2. First three language pairs En-{Fr,Zh,Cs} are high-resource.

Empirical Validation of Theorem 2   We first validate that Temperature Sampling has a lower variance in gradient estimation than scalarization, as predicted by Theorem 2. Figure 3 illustrates the variance between mini-batch gradients for TS and scalarization, confirming that TS reduces gradient variance ($2.25 \rightarrow 0.62$).
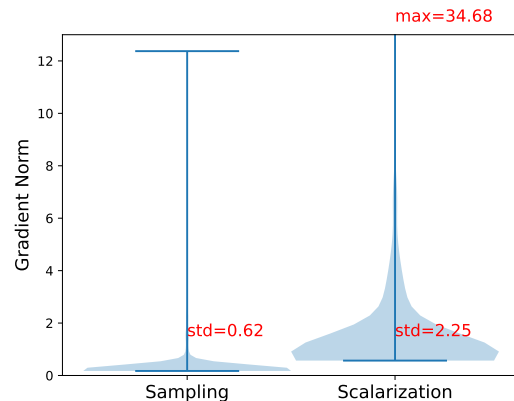

Figure 3: The distribution of gradient norm between mini-batches on En-{Cs, Ro} for Temperature Sampling and Scalarization. **Scalarization induces a larger variance ($2.25 > 0.62$) between mini-batch gradient norms compared to Temperature Sampling, as indicated by Theorem 3.**

Next, we observe that this lower variance leads to faster convergence during training. Figure 4 shows the validation loss curves over training iterations, where TS consistently converges faster than scalarization across all temperatures we experimented with. Notably, the larger the temperature, the greater the gap in convergence speed between TS and Scalarization. This suggests that when

---

[1]We apply the Adam (Kingma and Ba, 2015) optimizer with mini-batch gradient descent.

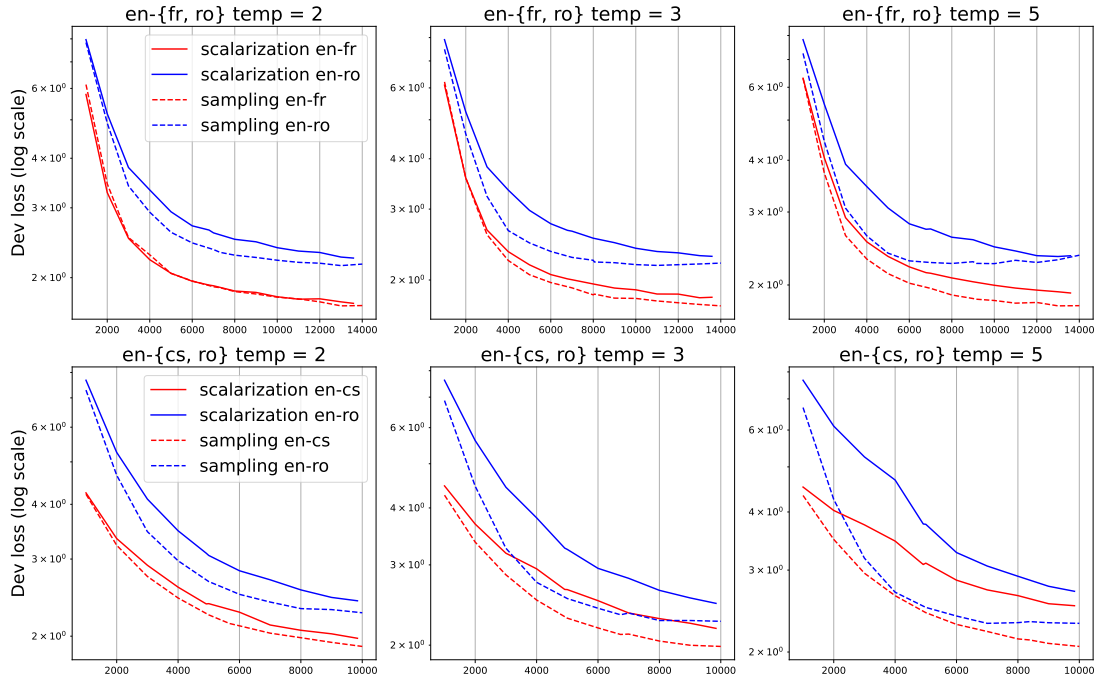[2]Additional experiments on En-{Zh, Ro} can be found in Appendix E.

Figure 4: Validation loss by training iteration for En-{Cs, Ro} (first row) and En-{Fr, Ro} (second row). **Temperature Sampling (dashed) converges faster compared to Scalarization (solid), leading to better performance on both the** HRL **and the** LRL**.**

there is a significant mismatch in data sizes, using Temperature Sampling with a high temperature is beneficial. Intuitively, low-resource languages (LRLs) with very little data should be upsampled more aggressively to accelerate convergence. We summarize our findings below:

Large Temperature Sampling is prone to overfitting In our En - {Fr, Ro} experiments, we observed that the model overfits the low-resource direction (Ro) when using a large temperature ($\tau = 3, 5$). However, the high-resource direction has not yet converged when this overfitting occurs; therefore, continuing to train in both directions would lead to severe overfitting. This indicates that we need to pair strong regularization on the LRL (e.g. early stopping) with a large temperature, which motivates our temperature scheduling method COOLDOWN, which uses a large temperature during the beginning to speed up training and then decreases the temperature to prevent overfitting on the LRL.

Temperature Sampling is equivalent to Scalarization given enough compute We found that Temperature Sampling (dashed in Figure 4) always converges faster compared to weighting the losses of individual directions (solid in Figure 4), but they eventually converge to the same validation

loss given enough training iterations, which corresponds to their equivalency under *full* gradient descent (Theorem 1). This means both Scalarization and Temperature Sampling can effectively balance multiple languages when given a large enough compute.

4 COOLDOWN: Balanced Training for Heavily Imbalanced Datasets

Based on the theoretical analysis in §3.1 and the empirical results in §3.2, we conclude that **Temperature Sampling with a large temperature converges faster than Scalarization with equivalent weights but is more prone to overfitting on the LRL when using a large temperature.** We thus hypothesize that we can employ a large temperature during the beginning of training to speed up convergence and then decrease the temperature to prevent overfitting on low-resource directions.

Our proposed method: We design a simple temperature scheduling method: COOLDOWN which starts with a high temperature (aggressive upsampling of LRLs) and then lowers the temperature to $\tau = 1$ (proportional sampling) at a fixed iteration to prevent overfitting. We describe our experiment setup at §4.1, report results at §4.2, and discuss our findings at §4.3.

3329

4.1 Setup

Models, Datasets, and Hyper-parameters We experiment on two setups: **multilingual machine translation** and **multilingual language modeling**, both suffering severe mismatch in dataset sizes. For our machine translation experiments, we use the standard encoder-decoder Transformer (Vaswani et al., 2017) architecture implemented in fairseq (Ott et al., 2019). We select 8 distinct languages from the opus-100 dataset (Zhang et al., 2020) and train a one-to-many translation where the source language is English, we used a shared BPE tokenizer with 64k vocabulary. Detailed Languages and their respective sizes can be found in Table 3. For our multilingual language modeling experiments, we use a decoder-only Transformer model from Huggingface (Wolf et al., 2020) and select 4 linguistically diverse languages with varying amounts of data from the mC4 (Xue et al., 2021) dataset. The statistics are in Table 2. We used the mT5 tokenizer (Xue et al., 2021) for our experiments on mC4.

Languages	Tokens	mT5 weights $\tau = 3.33$
EN - English	2733B	5.67%
IT - Italian	162B	2.43%
ZH - Chinese	39B	1.67%
SW - Swahili	1B	0.5%

Table 2: Dataset statistics of our selected subset of C4. mT5 (Xue et al., 2021) effectively uses an sampling temperature of $\tau = 3.33$ to oversample LRLs.

For our machine translation experiments, we use $\tau = 5$ for the first 30k of training iterations and $\tau = 1$ for the second 30k. For our language modeling experiments, we use $\tau = 5$ for the first 50k training iterations and $\tau = 1$ for the second 50k. Detailed hyper-parameters are in Appendix D.

Baselines We experiment on baselines that apply a fixed temperature throughout training (static temperature) and baselines that adjust the temperature during training (dynamic temperature). For static temperature, we vary the sampling temperature in $\tau = 1$ (Proportional Sampling), $\tau = 5$ and $\tau = 100$ ($\sim$Uniform Sampling). For dynamic temperature, we compare with **Unimax** (Chung et al., 2023), which first heavily upsamples low-resource languages and removes them after the low-resource dataset has been seen by the model for a fixed amount of repetitions, and **Order Matters** (Choi et al., 2023) which first only trains on high-resource languages, and only adds in low-resource languages to the end of training. Additionally, we include the results of **DoReMi** (Xie et al., 2023), which trains small proxy models that minimize the loss of a worse performing set of domains iteratively to find optimal sampling probabilities of each domain for training a large model.

4.2 Main Results

Machine Translation Table 3 shows applying COOLDOWN on training a multilingual machine translation model. Compared to proportional sampling ($\tau = 1$), COOLDOWN is able to greatly improve the mid and low-resource languages (+0.7 and 3.1 BLEU) while minimally sacrificing the performance of high-resource languages (-0.2 BLEU). Compared to static up-sampling $\tau = 5$, COOLDOWN matches the performance of mid- and low-resource languages while improving the performance of high-resource languages by 1.5 BLEU. Our method also outperforms the performance of *Unimax* and *Order Matters* scheduling while being easier to implement. Furthermore, COOLDOWN is able to match the performance of *DoReMi* without having to train multiple proxy models.

Multilingual Language Modeling We also experimented with the general language modeling task on multiple languages on selected languages on the multilingual C4 (mC4) dataset (Xue et al., 2021). We report the validation loss in table 4. Our results echo the findings in our machine translation experiments: COOLDOWN matches the performance on the only HRL English (EN) but outperforms other baselines on all three other languages.

4.3 Study on Temperature Schedules

In this section, we revisit two design choices in dynamic Temperature Sampling: 1) Increasing v.s. Decreasing the temperature during training, and 2) Dense v.s. Sparse updates of the temperatures. We highlight our results below:

Increasing the temperature is better than decreasing the temperature. *Unimax* (Chung et al., 2023) upsamples the LRLs during the **beginning** of training while *Order Matters* (Choi et al., 2023) upsamples LRLs at the **end** of training. To resolve this conflict, we compare an increasing temperature schedule (1 for the first 15k training iterations and 5 for the second 15k training iterations; "1-5")

en-{} # of Parallel Sentences	es 1M	fa 1M	ga 294K	gl 519K	ha 102K	hi 538K	it 1M	kk 83K	ug 76K	HRLs >1M	LRLs <500K
Static Temperature Sampling											
$\tau = 1$ (Proportional)	**38.9**	13.1	58	**28.9**	41.9	17.1	**32.8**	22.4	10.8	**28.3**	30.4
$\tau = 5$	36.9	12.2	**60.9**	28.3	46.3	16.8	30.7	26.8	9.4	26.6	**32.4**
$\tau = 100$ ($\sim$Uniform)	36.1	12.3	59.6	27.3	46.4	17.3	30.3	27.6	9.1	26.2	31.4
Dynamic Temperature Sampling											
Unimax (Chung et al., 2023)	37.1	12.2	60.7	28.4	45.9	16.5	30.8	26.2	9.4	26.7	32.1
Order Matters (Choi et al., 2023)	37.1	12.2	**60.9**	28.2	46.1	16.7	30.8	26.7	9.4	26.7	32.3
COOLDOWN (Ours)	38.7	**13.2**	60.1	28.7	**47.4**	17.3	32.2	26	11	28.1	**32.4**
With Proxy Model Training											
DoReMi (Xie et al., 2023)	37.3	13.1	60.4	28.4	46.3	17.1	29.8	23.0	10.8	26.7	**32.4**

Table 3: SacreBLEU scores (higher is better) on a chosen subset of OPUS-100 with a mixture of high (1M), mid (500K - 1M), and low (<500K) resource languages. The best performance is **bolded**. Scores that are close (within 1 BLEU) of the best performance are colored in green. Scores lower than the best for more than 1.5 BLEU are highlighted in red. **COOLDOWN outperforms various static and dynamic Temperature Sampling methods,** by improving the performance on LRLs without sacrificing much performance on HRLs.

Method	EN 2733B	IT 162B	ZH 39B	SW 1B
Static Temperature Sampling				
$\tau = 1$ (Proportional)	**2.67**	3.10	4.23	4.09
$\tau = 5$	2.91	3.01	3.14	3.12
$\tau = 100$ ($\sim$Uniform)	3.02	2.88	2.76	3.01
Dynamic Temperature Sampling				
Order Matters (Choi et al., 2023)	2.77	2.76	3.23	3.26
Unimax (Chung et al., 2023)	2.98	2.91	2.85	3.06
COOLDOWN (Ours)	2.75	**2.63**	2.56	2.94
With Proxy Model Training				
DoReMi (Xie et al., 2023)	2.89	**2.63**	**2.51**	**2.89**

Table 4: Dev loss on selected mC4 subset (lower is better). **COOLDOWN achieves the best performance on all three LRLs (IT, ZH, and SW).** On the only HRL English, COOLDOWN is only behind proportional sampling (τ=1), which is heavily optimized towards performance on English by sacrificing performance on all other languages.

with a decreasing temperature schedule (5 for the first 15k iterations and 1 for the second half; "5-1"), using the same En-{Zh, Ro} machine translation data in §3.2. Figure 8 illustrates that a decreasing schedule (dashed) converges faster and results in better performance on the LRL compared to an increasing schedule (solid), with minimal sacrifice on the HRL. This means that upsampling the LRLs during the beginning of training performs better.

Curse of Granularity We concluded that a decreasing schedule generally leads to faster convergence and better overall performance. Furthermore, we compare various fine-grained decreasing schedules that perform dense decreasing of the sampling

temperature with our sparse update (5 for the first 50k, 1 for the second 50k training iterations) using the subset in mC4 described in Table 2. Figure 5 illustrates the schedules we compare, and 6 shows the validation loss of each decreasing schedule. Fine-grained online reweighting method yields worse performance, echoing the findings in Fan et al. (2024) and Xie et al. (2023).[3]

---

[3]Both Xie et al. (2023) and Fan et al. (2024) approximate sampling probabilities using Scalarization when performing dense updates, which could also be the reason why dense update underperforms in their experiments.
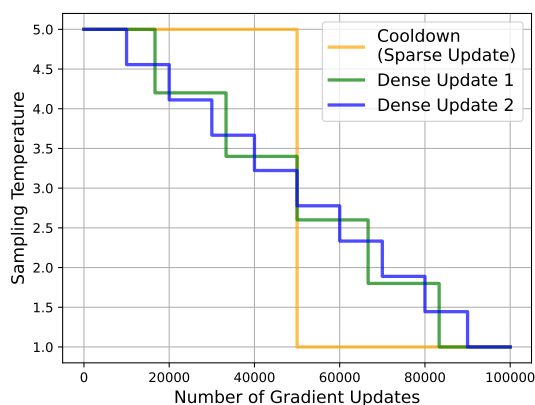
Figure 5: Sampling temperature schedules.

Method	EN	IT	ZH	SW
COOLDOWN	**2.59**	**2.63**	**2.56**	**2.94**
Dense Update 1	3.17	2.89	2.85	3.11
Dense Update 2	3.31	3.02	3.35	3.16

Figure 6: Validation loss on mC4 subset (lower is better) for different update schedules in Figure 5. Sparse updates perform better than dense updates.

5 Related Works

Gradient-based methods for Multi-Task Learning Training a multi-domain language model can be seen as Multi-Task Learning (MTL; Caruana (1997)), where each domain is a single task. Gradient-based methods aim to reduce the discrepancy in directions of conflicting gradients in different tasks: *PCGrad* (Yu et al., 2020) aims to project the gradient of a task onto the orthogonal plane of the gradients of the other task. Another line of work (Wang et al., 2020a,b; Kreutzer et al., 2021) uses gradient similarity as the reward to train a policy that decides the sampling probabilities for each domain in a Reinforcement Learning setting. Fan et al. (2024) utilizes gradient similarity between domains to design a temperature schedule for balancing multiple domains in training language models. However, recent studies point out that such gradient-based techniques do not yield significant improvement compared to a weighted sum of individual task losses (Scalarization) (Kurin et al., 2022; Xin et al., 2022; Royer et al., 2023). Our results also echo the findings of Zhai et al. (2023), where they find loss reweighting (Scalarization) underperforms standard training. Our work provides
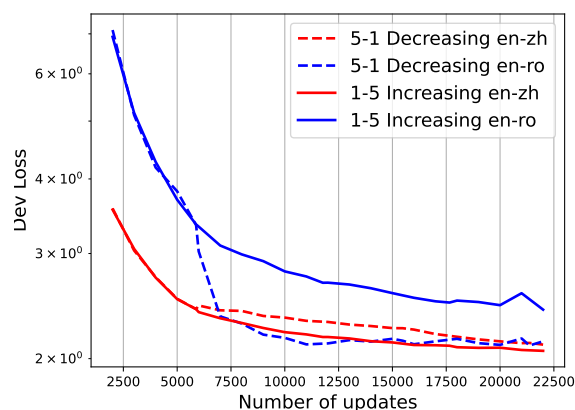

Figure 7: Comparison between an increasing (solid) Temperature Sampling schedule and a decreasing (dashed) schedule in multilingual machine translation En-{Zh, Ro}. **A decreasing temperature schedule outperforms an increasing one on the LRL with minimal sacrifice on the HRL.**

a possible explanation that Scalarization induces a larger variance in gradients.

Loss-based methods for Multi-Task Learning Another line of work utilizes the loss instead of the gradients per task for optimizing MTL models. An intuitive method is to put more weight on the task with the highest loss. In statistical learning, Distributionally Robust Optimization (DRO) methods (Ben-Tal et al., 2011; Duchi and Namkoong, 2021; Hashimoto et al., 2018; Sagawa* et al., 2020) minimize the loss of the worst-performing subgroup to balance performance. Oren et al. (2019) and Xie et al. (2023) apply DRO to multi-domain language modeling to minimize the loss of a set of worse-performing domains. Similarly, Zhou et al. (2021) applies DRO to multilingual machine translation by minimizing the loss of a set of worse-performing translation directions. COOLDOWN can be seen as an efficient approximation of DRO methods by up-sampling the worse-performing LRL and shifting the focus to the HRL once the LRL is sufficiently trained. Unlike DRO, COOLDOWN do does not require training proxy models (Liu et al., 2021) and dense updates on domain weights. Our findings also connect to the fact that different languages act like regularizers in multi-task learning (Li and Murray, 2023).

We defer additional related works on addressing multilingual imbalance and the discussion between class imbalance and domain imbalance to Appendix A.

Scaling Laws for Domain Mixture The search space for the optimal domain weights at any given training iteration is combinatorially large. Existing works conduct comprehensive experiments on smaller scaled models to learn how the training and generalization error varies according to dataset sizes and domain weights — "scaling laws" of domain mixture (Ye et al., 2024; Ge et al., 2024; Jiang et al., 2024). Closer to our work, Chen et al. (2023) fits scaling laws for sampling temperature for multilingual machine translation. Concurrent to our work, He et al. (2024) fits scaling laws for multilingual language modeling. However, as Jiang et al. (2024) denotes: "the optimal data policy for a smaller model does not necessarily generalize to larger models."

6 Conclusion

We examined two common balancing methods for multi-domain language modeling with data imbalances: Scalarization and Temperature Sampling. Although both yield the same loss, Temperature Sampling converges faster but risks overfitting. To mitigate this, we propose COOLDOWN, a variant that adjusts temperatures during training to maintain fast convergence while reducing overfitting.

Limitations

We discuss the limitations of our study here.

Impact of data mixture on downstream performance Studies have pointed out that the data mixtures in different domains impact downstream performance (Gururangan et al., 2020; Albalak et al., 2023; Fan et al., 2024). Our work only focuses on the impact of different temperature schedules on pre-training validation performance. Although the effect of optimizing pre-training mixtures across different languages on downstream performances has not been fully concluded, works have shown that a lower pre-train validation loss generally leads to better downstream performance (Xie et al., 2023; Du et al., 2024).

Difference between multi-lingual and multi-domain language modeling Existing work on mono-lingual language modeling (Chowdhery et al., 2023; Du et al., 2022; Xie et al., 2023; Oren et al., 2019; Fan et al., 2024; Longpre et al., 2023) maps all data from the same source (e.g. Wikipedia, Web, Books) to a single domain. Such a mapping ignores the subdomains within each source. In our work, we focused on the multilingual setup because (a) There exists a severe data size mismatch and (b) there is a clear and natural definition of "domain" — the different languages, and we expect the sampling to have a larger impact because of this mismatch. Even though we only conducted experiments on a multilingual setup, our theoretical analysis applies to all setups with heavy dataset size mismatches.

Finding the optimal temperature schedule It requires a large amount of compute to thoroughly study the scaling laws of pre-training language models under different temperatures (Ye et al., 2024) and to search for the optimal static sampling temperature τ for a given dataset (Chen et al., 2023), let along dynamic temperature scheduling. The optimal temperature depends not just on the size of the dataset but also on the "difficulty" of the dataset. Therefore, existing research (Chen et al., 2023; Xie et al., 2023; Oren et al., 2019; Zhou et al., 2021; Liu et al., 2024; Dubey et al., 2024) relies on training a proxy model on the dataset probe domain difficulty to determine optimal weights. Our work, instead, proposes a heuristic that decreases the temperature during training that does not rely on any proxy training, but we did not exhaustively test all the decreasing schedules. We leave finding optimal temperature schedules without preliminary training for future work.

Ethical Considerations

One application of our study is to balance high- and low-resource languages. We aim to mitigate biases that favor high-resource languages. However, this approach also raises risks, such as amplifying existing biases in limited and potentially skewed low-resource corpora. Furthermore, improved models could be misused to spread misinformation or infringe on privacy, especially in communities less equipped to counter such impacts. Thus, while COOLDOWN promotes linguistic diversity, it requires careful monitoring to ensure it is used ethically.

Acknowledgements

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. 2023. Efficient online data mixing for language model pre-training. *Preprint*, arXiv:2312.02406.

Aharon Ben-Tal, Dick den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. 2011. Robust solutions of optimization problems affected by uncertain probabilities. *Advanced Risk & Portfolio Management® Research Paper Series*.

Mateusz Buda, Atsuto Maki, and Maciej Mazurowski. 2017. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106.

Rich Caruana. 1997. Multitask learning. *Machine Learning*.

Liang Chen, Shuming Ma, Dongdong Zhang, Furu Wei, and Baobao Chang. 2023. On the pareto front of multilingual neural machine translation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Dami Choi, Derrick Xin, Hamid Dadkhahi, Justin Gilmer, Ankush Garg, Orhan Firat, Chih-Kuan Yeh, Andrew M. Dai, and Behrooz Ghorbani. 2023. Order matters in the presence of dataset imbalance for multilingual learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Xavier Garcia, Adam Roberts, Yi Tay, Orhan Firat, Sharan Narang, and Noah Constant. 2023. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. In *The Eleventh International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.

Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. 2024. Understanding emergent abilities of language models from the loss perspective. *Preprint*, arXiv:2403.15796.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock,

Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li,

Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook

Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

John C. Duchi and Hongseok Namkoong. 2021. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378 – 1406.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2024. Doge: Domain reweighting with generalization estimation. In *The Forty-first International Conference on Machine Learning*.

Vitaly Feldman. 2020. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 954–959, New York, NY, USA. Association for Computing Machinery.

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: discovering the long tail via influence estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Ce Ge, Zhijian Ma, Daoyuan Chen, Yaliang Li, and Bolin Ding. 2024. Bimix: Bivariate data mixing law for language model pretraining. *Preprint*, arXiv:2405.14908.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1929–1938. PMLR.

Yifei He, Alon Benhaim, Barun Patra, Praneetha Vaddamanu, Sanchit Ahuja, Parul Chopra, Vishrav Chaudhary, Han Zhao, and Xia Song. 2024. Scaling laws for multilingual language models. *Preprint*, arXiv:2410.12883.

Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2023. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 523–540, Dubrovnik, Croatia. Association for Computational Linguistics.

Yichong Huang, Xiaocheng Feng, Xinwei Geng, Baohang Li, and Bing Qin. 2023. Towards higher Pareto frontier in multilingual machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3802–3818, Toronto, Canada. Association for Computational Linguistics.

Yichong Huang, Xiaocheng Feng, Xinwei Geng, and Bing Qin. 2022. Unifying the convergences in multilingual neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6822–6835, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yiding Jiang, Allan Zhou, Zhili Feng, Sadhika Malladi, and J. Zico Kolter. 2024. Adaptive data optimization: Dynamic sample selection with scaling laws. *Preprint*, arXiv:2410.11820.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Julia Kreutzer, David Vilar, and Artem Sokolov. 2021. Bandits don't follow rules: Balancing multi-facet machine translation with multi-armed bandits. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3190–3204, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and M. Pawan Kumar. 2022. In defense of the unitary scalarization for deep multi-task learning. In *Advances in Neural Information Processing Systems*.

Tianjian Li and Kenton Murray. 2023. Why does zero-shot cross-lingual generation fail? an explanation and a solution. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12461–12476, Toronto, Canada. Association for Computational Linguistics.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR.

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

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2023. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. *Preprint*, arXiv:2305.13169.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. 2018. An empirical model of large-batch training. *Preprint*, arXiv:1812.06162.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Herbert E. Robbins. 1951. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.

Amelie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. 2023. Scalarization for multi-task and multi-domain learning at scale. In *Advances in Neural Information Processing Systems*, volume 36, pages 16917–16941. Curran Associates, Inc.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *International Conference on Learning Representations*.

Uri Shaham, Maha Elbayad, Vedanuj Goswami, Omer Levy, and Shruti Bhosale. 2023. Causes and cures for interference in multilingual translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15849–15863, Toronto, Canada. Association for Computational Linguistics.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1139–1147, Atlanta, Georgia, USA. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Jaime Carbonell, and Graham Neubig. 2020a. Optimizing data usage via differentiable rewards. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020b. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2021. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin

Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. 2022. Do current multi-task optimization methods in deep learning even help? In *Advances in Neural Information Processing Systems*, volume 35, pages 13597–13609. Curran Associates, Inc.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. 2024. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *Preprint*, arXiv:2403.16952.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc.

Runtian Zhai, Chen Dan, J Zico Kolter, and Pradeep Kumar Ravikumar. 2023. Understanding why generalized reweighting does not improve over ERM. In *The Eleventh International Conference on Learning Representations*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Chunting Zhou, Daniel Levy, Xian Li, Marjan Ghazvininejad, and Graham Neubig. 2021. Distributionally robust multilingual machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5664–5674, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Supplemental Material

A Additional Related Works

Multilingual Interference   Finding out the reasons and solutions for negative interference in Multilingual Neural Machine Translation (Johnson et al., 2017; Aharoni et al., 2019) has been an active research area for the past decade. Yet, while previous studies (Wang et al., 2021) find that negative interference mainly occurs between different language families, recent studies (Shaham et al., 2023) have demonstrated that negative inference does not happen between languages of different families. The interference emerges because of the mismatch in the amount of data for different translation directions. Real-world translation data suffers from a heavy mismatch of data size in different directions, ranging from less than 100K to over 100M (NLLB Team et al., 2022). In our work, we show that this heavy mismatch in data size results in low-resource languages being under-trained.

To mitigate interference caused by dataset sizes, Aharoni et al. (2019) and Xue et al. (2021) propose to up-sample low-resource languages, which often results in the model overfitting on the LRLs while underfitting HRLs. Huang et al. (2022) proposes to distill the model from earlier checkpoints with the LRLs that have not overfit with the current model to regularize the training of LRLs. Huang et al. (2023) proposes to distill between a model trained with a low sampling temperature and a model trained with a high sampling temperature. *Unimax* (Chung et al., 2023) proposes to first uniformly sample from all languages until an LRL dataset has been seen by the model for a fixed amount of repetitions; then, we remove the LRL from training. *Order-Matters* (Choi et al., 2023) proposes the opposite of *Unimax* (Chung et al., 2023), to first only train on the HRL and add in the LRL after a fixed iteration. Our work shows that the *Unimax* style of decreasing the temperature works better and proposes a simple alternative that does not require tracking how many times a model has seen an individual LRL dataset.

Class Imbalance v.s. Domain Imbalance   Class imbalance aims to address when the input x is drawn from the same distribution $x \sim \mathcal{D}$ but the output labels y are imbalanced. Domain imbalance, on the other hand, studies the problem when the input is drawn from different distributions with mismatched sizes $x \sim \{\mathcal{D}_1, \mathcal{D}_2, ...\mathcal{D}_k\}$, making no assumptions about the output labels y. Therefore, our study is distantly connected to adjusting the sampling probabilities to address class imbalance (Buda et al., 2017). We refer readers to Henning et al. (2023) for a comprehensive survey of class imbalance in natural language processing.

B Proof of Theorem 2

Theorem 2 (Scalarization induces larger variance under Stochastic Gradient Descent) *Using the same notation in Corollary 1.1, we have* $\mathrm{Var}(\nabla \mathcal{L}_S(x; \mathbf{w}_\tau)) \geq \mathrm{Var}(\nabla \mathcal{L}_{TS}(x; \tau))$.

We first proof a required lemma:

Lemma 2.1 Let $\mathcal{D} = \{\mathcal{D}_1, ..., \mathcal{D}_K\}, \forall j \in \{1, ..., K\}$, let $p(j, \tau) = \dfrac{\mathcal{D}_j^{\frac{1}{\tau}}}{\sum_k \mathcal{D}_k^{\frac{1}{\tau}}}$, then $\sum_{i=1}^K \frac{p(i;\tau)^2}{p(i;1)} \geq 1$ holds true when $\forall j, \mathcal{D}_j \geq 0$.

Proof. We substitute $x_j = \mathcal{D}_j^{\frac{1}{\tau}}$, then:

$$\sum_{i=1}^K \frac{p(i;\tau)^2}{p(i;1)} = \frac{\sum_i x_i^\tau}{(\sum_i x_i)^2} \cdot \left(\sum_i x_i^{2-\tau} \right)$$

By the Cauchy-Schwartz inequality, we have:

$$\left(\sum_i x_i^{\tau/2} \cdot x_i^{(2-\tau)/2} \right)^2 \leq \left(\sum_i x_i^\tau \right) \left(\sum_i x_i^{2-\tau} \right),$$

3339

which simplifies to:

$$\left(\sum_i x_i\right)^2 \leq \left(\sum_i x_i^\tau\right)\left(\sum_i x_i^{2-\tau}\right),$$

which implies that:

$$1 \leq \frac{\sum_i x_i^\tau}{(\sum_i x_i)^2} \cdot \left(\sum_i x_i^{2-\tau}\right) = \sum_{i=1}^K \frac{p(i;\tau)^2}{p(i;1)}.$$

$\square$

Armed with Lemma 2.1, we come back to the proof of Theorem 2.

Proof. Since $\nabla\mathcal{L}_S(x; \mathbf{w}_\tau)$ and $\nabla\mathcal{L}_{TS}(x; \tau)$ are unbiased estimates of the total gradient, we only need to show that the expectation of the *squared* gradient is larger for the stochastic gradient under Scalarization.

$$\underbrace{\mathbb{E}_{x\sim\mathcal{D}}[(w_{f(x)}\nabla\mathcal{L}(x))^2]}_{\text{Scalarization}} = \sum_{i=1}^K p(i;1)w_{f(x)}^2\nabla\mathcal{L}^2(x)$$

$$\sum_{i=1}^K \frac{p(i;\tau)^2}{p(i;1)}\nabla\mathcal{L}^2(x) \geq \nabla\mathcal{L}^2(x)\sum_{i=1}^K p(i;\tau)\nabla\mathcal{L}^2(x)$$

$$= \underbrace{\mathbb{E}_{\mathcal{D}_i\sim p(i;\tau)}\left[\mathbb{E}_{x\sim\mathcal{D}_i}[\nabla\mathcal{L}^2(x)]\right]}_{\text{Temperature Sampling}}.$$

$\square$

C   Proof of Theorem 3 and Construction of Figure 2

Theorem 3 (Scalarization induces larger variance when approximating higher temperatures)
The difference in variance $\Delta = \text{Var}(\nabla\mathcal{L}_S(x)) - \text{Var}(\nabla\mathcal{L}_{TS}(x))$ *is non-decreasing when* $\tau \geq 1$.

Proof. From the proof of Theorem 2, we know that the difference in variance Δ can be quantified by $\frac{p(\mathcal{D}_i;\tau)^2}{p(\mathcal{D}_i;1)}$. We substitute $x_j = \mathcal{D}_j^{\frac{1}{\tau}}$, then:

$$\sum_{i=1}^K \frac{p(i;\tau)^2}{p(i;1)} = \frac{\sum_i x_i^\tau}{(\sum_i x_i)^2} \cdot \left(\sum_i x_i^{2-\tau}\right).$$

Let $F(\tau) = \frac{\sum_i x_i^\tau}{(\sum_i x_i)^2} \cdot \left(\sum_i x_i^{2-\tau}\right)$ be a function of τ. Taking its derivative with respect to τ:

$$\frac{dF(\tau)}{d\tau} = F(\tau) \times \left[\left(\sum_i \frac{x_i^\tau \log x_i}{\sum_j x_j^\tau}\right) - \left(\sum_i \frac{x_i^{2-\tau} \log x_i}{\sum_j x_j^{2-\tau}}\right)\right].$$

Taking the derivative of the term $\sum_i \frac{x_i^\tau \log x_i}{\sum_j x_j^\tau}$ with respect to τ, we get:

$$\frac{d}{d\tau}\sum_i\left(\frac{x_i^\tau \log x_i}{\sum_j x_j^\tau}\right) = \frac{\sum_j x_j^\tau \sum_i x_i^\tau (\log x_i)^2 - (\sum_i x_i^\tau \log x_i)^2}{(\sum_i x_i^\tau)^2},$$

which is the variance of $\log x_i$ under the probability distributions $\frac{x_i^\tau}{\sum_i x_i^\tau}$. Similarly, the derivative of $\sum_i \frac{x_i^{2-\tau} \log x_i}{\sum_j x_j^{2-\tau}}$ with respect to τ is the negative variance of $\log x_i$ under distribution $\frac{x_i^{2-\tau}}{\sum_i x_i^{2-\tau}}$. Since variances are always non-negative, we conclude that the following difference:

3340

$$\left[\left(\sum_i \frac{x_i^\tau \log x_i}{\sum_j x_j^\tau}\right) - \left(\sum_i \frac{x_i^{2-\tau} \log x_i}{\sum_j x_j^{2-\tau}}\right)\right],$$

is always non-negative. By Lemma 2.1, we know that $F(\tau) \geq 1$. Therefore, the derivative $\frac{dF(\tau)}{d\tau}$ is always non-negative when $\tau \geq 1$, meaning that $F(\tau)$ is non-decreasing when $\tau \geq 1$.

Furthermore, when τ is strictly larger than 1 and not all $\mathcal{D}$ are equal, $F(\tau)$ monotonically increases with τ. Showing that approximating a larger temperature using Scalarization induces a larger variance than approximating smaller temperatures. □

Construction of Figure 2 We plot the function $F(\tau) = \sum_i \frac{p(i;\tau)^2}{p(D_i;1)}$ against τ by starting with a uniform distribution and progressively increasing its skewness to resemble Zipf distributions. Specifically, we generate distributions $D_i \propto \frac{1}{i^\alpha}$ for various exponents α (ranging from 0 to higher values), where i denotes the rank of each element. For each distribution, we compute the normalized probabilities $p(i;\tau) = \frac{|D_i|^{1/\tau}}{\sum_j |\mathcal{D}_j|^{1/\tau}}$ across a range of τ values. This approach allows us to analyze how increasing the skewness of the distribution influences the behavior of $F(\tau)$ as a function of τ.

D Detailed Hyper-Parameters

We provide a comprehensive list of the hyper-parameters we used in this appendix section: §3.2 - Table 5.

Hyper-parameter	Value	Hyper-parameter	Value
Arch	wmt_en_de_big	Label smoothing	0.1
Optimizer	adam	Adam epsilon	1e-06
Adam betas	"(0.9, 0.98)"	Learning rate scheduler	inverse_sqrt
Learning rate	0.0005	Warmup updates	4000
Validate interval updates	1000	Dropout	0.1
Attention dropout	0.1	Weight decay	0.0
Max tokens	32768	Update frequency	8
Max source positions	256	Max target positions	256

Table 5: Detailed Hyper-parameters for experiments in §3.2. We use the fairseq (Ott et al., 2019) implementation.

Hyper-parameter	Value	Hyper-parameter	Value
Arch	iwslt_de_en	Label smoothing	0.1
Optimizer	adam	Adam epsilon	1e-06
Adam betas	"(0.9, 0.98)"	Learning rate scheduler	inverse_sqrt
Learning rate	0.0005	Warmup updates	4000
Validate interval updates	1000	Dropout	0.1
Attention dropout	0.1	Weight decay	0.0
Max tokens	16384	Update frequency	4
Max source positions	256	Max target positions	256

Table 6: Detailed Hyper-parameters for Machine Translation experiments in §4 on our selected subset of opus-100 (Zhang et al., 2020). We use the fairseq (Ott et al., 2019) implementation. Differences against Table 5 are in red.

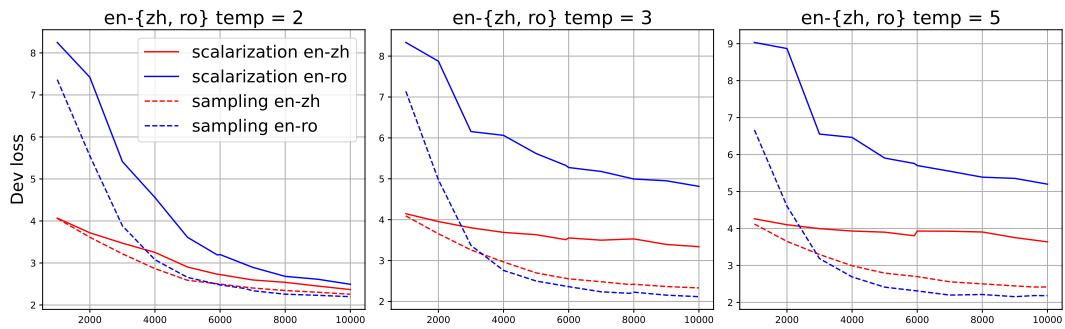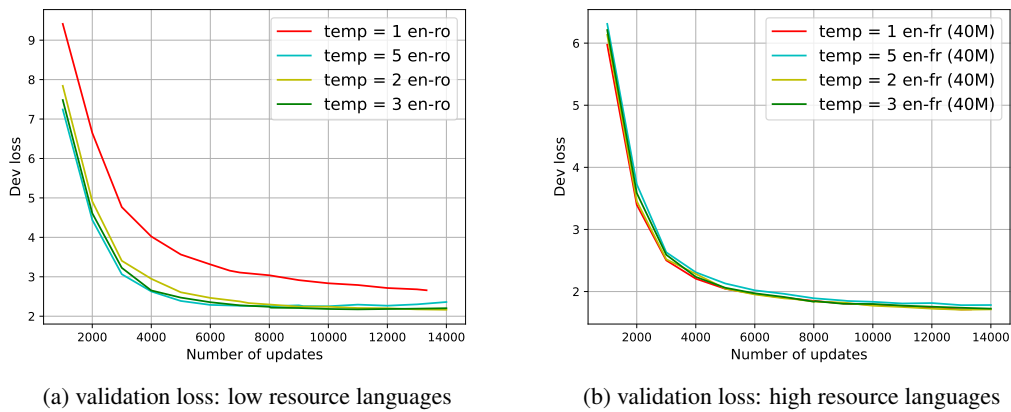# E Additional Results on Scalarization V.S. Temperature Sampling

Figure 8: Validation loss by training iteration for En-{Zh, Ro}. Temperature Sampling (Dashed) converges much faster than Scalarization (Solid), especially at higher temperatures.

(a) validation loss: low resource languages

(b) validation loss: high resource languages

Figure 9: Validation loss by gradient updates on the low-resource and high-resource language (left) jointly trained on the same model. Adjusting the sampling temperature has little impact on the high-resource language but a high impact on the low-resource language.