Towards Higher Pareto Frontier in Multilingual Machine Translation

Yichong Huang[†], **Xiaocheng Feng**[†‡], **Xinwei Geng**[†], **Baohang Li**[†], **Bing Qin**[†‡]
[†]Harbin Institute of Technology [‡] Peng Cheng Laboratory
{ychuang,xcfeng,xwgeng,baohangli,qinb}@ir.hit.edu.cn

Abstract

Multilingual neural machine translation has witnessed remarkable progress in recent years. However, the long-tailed distribution of multilingual corpora poses a challenge of Pareto optimization, *i.e.,* optimizing for some languages may come at the cost of degrading the performance of others. Existing balancing training strategies are equivalent to a series of Pareto optimal solutions, which trade off on a Pareto frontier[1]. In this work, we propose a new training framework, Pareto Mutual Distillation (Pareto-MD), towards pushing the Pareto frontier outwards rather than making trade-offs. Specifically, Pareto-MD collaboratively trains two Pareto optimal solutions that favor different languages and allows them to learn from the strengths of each other via knowledge distillation. Furthermore, we introduce a novel strategy to enable stronger communication between Pareto optimal solutions and broaden the applicability of our approach. Experimental results on the widely-used WMT and TED datasets show that our method significantly pushes the Pareto frontier and outperforms baselines by up to +2.46 BLEU[2].

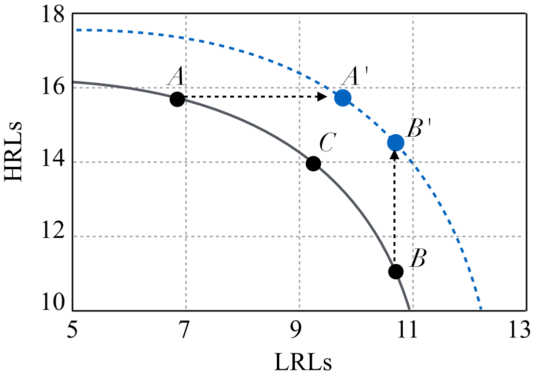


Figure 1: Multilingual performance frontier shifts outwards. X-axis and Y-axis indicate the performance of Low-Resource Languages and High-Resource Languages, respectively. Existing methods reflect a trade-off on the Pareto frontier (*i.e.,* the gray curve). Our work aims to *push the original Pareto frontier i.e.,* the blue dotted curve. To this effect, we ameliorate each individual model's shortcoming while retaining their strengths, *e.g.,* moving right the solution A to A' and moving up the solution B to B', via our Pareto Mutual Distillation.

1 Introduction

Multilingual neural machine translation (MNMT) is a popular paradigm that uses a unified model to handle the entire translation process for multiple language pairs (Ha et al., 2016; Firat et al., 2016; Johnson et al., 2017). This paradigm is particularly effective at improving the performance of low-resource languages through transfer learning (Aharoni et al., 2019; Dabre et al., 2020; Siddhant et al., 2022). Besides, MNMT is highly deployable since only one model is required (Fan et al., 2021; Yang et al., 2021; NLLB Team et al., 2022).

However, the severely imbalanced distribution of multilingual training data puts the MNMT in a situation of Pareto optimization (also known as multi-objective optimization). That is, when some languages are optimized, others degenerate. Existing methods can be considered a set of Pareto optimal solutions that trade off on a Pareto frontier, which focus on balancing the performance across different languages by adjusting the sampling distribution (Arivazhagan et al., 2019; Wang et al., 2020; Wu et al., 2021). The widely-used temperature-based sampling (Arivazhagan et al., 2019) is typical evidence of the claim above, which uses a hyper-parameter to smooth the training distribution over all language pairs to enhance the representation of low-source Languages (LRLs) while sacrificing the which of High-Resource Languages (HRLs). Despite the emergence of several sophis-

[1] In Pareto optimization, Pareto optimal solutions refer to solutions in which none of the objectives can be improved without sacrificing at least one of the other objectives. The set of all Pareto optimal solutions forms a Pareto frontier.

[2] Our code is publicly available at https://github.com/OrangeInSouth/Pareto-Mutual-Distillation

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3802–3818
July 9-14, 2023 ©2023 Association for Computational Linguistics

ticated dynamic sampling technologies designed to overcome the inflexibility of temperature-based sampling, their performance remains restricted to this Pareto frontier (Wang et al., 2020; Zhou et al., 2021; Zhang et al., 2021).

In this work, we propose a novel training framework, named Pareto Mutual Distillation (Pareto-MD), to push the Pareto frontier of multilingual models. Specifically, Pareto-MD uses different training distributions that favor dissimilar subsets of languages to train two multilingual models simultaneously. These two models learn from each other at each training step with knowledge distillation. The underlying idea of Pareto-MD is to address shortcomings of individual Pareto optimal solutions via access to a better one in terms of that shortcoming, thereby raising the Pareto frontier, as Fig. 1 depicts. To fully exploit the potential of our approach in multilingual settings, we further propose Automatic Pareto Mutual Distillation, which dynamically determines the contribution of distillation learning loss on each objective. These contributions, controlled by a set of distillation weights, adapt automatically to the evolving models, eliminating the need for manual hyper-parameter search.

While our method applies essentially to any multi-objective optimization problem, we specifically demonstrate its benefit on multilingual machine translation. The experimental results on two widely-used datasets demonstrate the effectiveness of our method, which improves up to +2.46 BLEU, and the further analysis shows the Pareto frontier is pushed outwards visibly.

2 Preliminaries

Neural machine translation (NMT) is a classic NLP task that translates a sentence x in source language into a sentence y in target language (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). Given a parallel corpus $D = \{(x, y) \in \mathcal{X} \times \mathcal{Y}\}$, the NMT model is commonly trained by minimizing the negative log-likelihood loss:

$$\mathcal{L}_{ce} = \sum_{(x,y)\sim D} \sum_{i\leqslant|y|} -\log p(y_i|x, y_{<i}; \theta), \quad (1)$$

where $p(\cdot|\cdot;\theta)$ maps the source sentence and previous generated text to the next target token.

2.1 Multilingual Machine Translation

Given a set of language pairs L, the MNMT model is trained on the combination of $|L|$ parallel datasets: $\{D_\ell^{train}\}_{\ell=1}^{|L|}$, where D_ℓ^{train} is the dataset of language pair (S_ℓ, T_ℓ). In order to encode and decode the text in various languages into and from a universal semantic space, a large multilingual vocabulary $\mathcal{V}$ is constructed. The language tag is appended to the beginning of source sentences as a hint of the target language. The MNMT model is also trained with the loss function as Eq.1 over the multilingual datasets.

Temperature-based Sampling. The multilingual datasets form a distribution P, where $P(\ell) = \frac{N_\ell}{\sum_j N_j}$ is the sampling probability of language pair ℓ and we denote the dataset size of D_ℓ^{train} by N_ℓ. Since sampling probabilities of LRLs are substantially lower than those of HRLs, the optimization towards LRLs can be overwhelmed by those of HRLs. To resolve this issue, Arivazhagan et al. (2019) propose temperature-based sampling, introducing a hyper-parameter τ to re-scale the smoothness of training distribution. Concretely, the sampling probability of each language pair ℓ is set to:

$$P(\ell) = \frac{N_\ell^{1/\tau}}{\sum_j N_j^{1/\tau}}, \quad (2)$$

increasing the value of τ produces smoother training distributions and stronger preferences on LRLs.

2.2 Mutual Distillation

Knowledge Distillation (KD) is a popular technology for knowledge transfer, which originates from compressing a static high-capacity model (teacher model) into a small compact model (student model) (Hinton et al., 2015). Mutual distillation is a variant of KD (Zhang et al., 2018; Guo et al., 2020). Instead of using a pre-trained teacher model, **mutual distillation involves training more than one model simultaneously**, with each model teaching the other throughout the training process. Mutual distillation takes the same loss function as vanilla knowledge distillation, that is:

$$\mathcal{L}_{kd} = \sum_{i\leqslant|y|} \sum_{w\in\mathcal{V}} -p(w|x, y_{<i}; \theta^T) \cdot \log p(w|x, y_{<i}; \theta^S), \quad (3)$$

where $\mathcal{V}$ is the target-side vocabulary, θ^S and θ^T are the student model and teacher model. *The major difference of Pareto-MD from vanilla mutual distillation is that we train two models with different sampling distributions to make them favor different sets of objectives.*

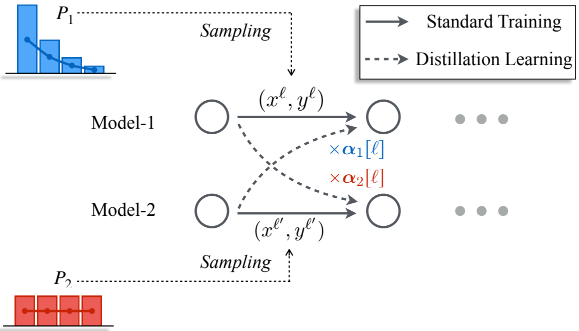


Figure 2: Illustration of Pareto-MD, using different sampling distributions to train two models. At each step, both models additionally mimic the output of each other via knowledge distillation. The distillation learning of each model is weighted by language-specific distillation weights $\boldsymbol{\alpha}_i[\ell]$ deduced with specific strategies.

Algorithm 1: Pareto-MD

```
Input      : Datasets $\{D_\ell^{train}\}_{\ell=1}^{|L|}$, two training
             distributions $P_1$, $P_2$, learning rate $\eta$,
             distillation weights updating strategy $\mathcal{S}$,
             updating interval $\mathcal{T}$
Initialize: Randomly initialize model $\theta_1$ and $\theta_2$, set
             multilingual distillation weights
             $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 = \mathbf{0}$, training step $t = 0$
while not converged do
    $t \leftarrow t + 1$
    for $i \leftarrow 1$ to 2 do
        Sample a language pair $\ell$ from $P_i$
        Draw a batch of samples $B_\ell$ from $D_\ell^{train}$
        $\theta_i \leftarrow \theta_i - \eta\nabla_{\theta_i}\mathcal{L}_{PMD}(B_\ell; \theta_i, \theta_{3-i}, \boldsymbol{\alpha}_i[\ell])$
    end
    if $t \,\%\, \mathcal{T} = 0$ then
        Update $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$ with the specific strategy $\mathcal{S}$
    end
end
```

3 Pareto Mutual Distillation

In this section, we first introduce our training framework Pareto-MD (§3.1). Next, two strategies that determine the important distillation weights, UNI-PMD and BI-PMD, are shown (§3.2). To overcome the flaws of these two strategies above, AUTO-PMD is further proposed (§3.3).

3.1 Framework

We illustrate our Pareto-MD in Fig. 2. Pareto-MD simultaneously trains two models, denoted by θ_1 and θ_2, using different sampling distributions, P_1 and P_2, that make each model favor a different set of language pairs. To obtain expected distributions, we adopt temperature-based sampling, as shown in Eq.2, and set $\tau = 1$ for P_1, $\tau > 1$ (*e.g.,* $\tau = 5$ commonly) for P_2. In this way, θ_1 prefers HRLs, and θ_2 prefers LRLs.

At each training step, for each model θ_i where $i \in \{1, 2\}$, Pareto-MD first draws a language pair ℓ from training distribution P_i, then a mini-batch of sentence pairs $B_\ell = \{x_\ell, y_\ell\}$ are sampled from D_ℓ^{train}. Next, the model θ_i is trained to fit B_ℓ and match the output of the other model, *i.e.,* θ_{3-i}. The overall loss function for model θ_i is defined as:

$$\begin{aligned}\mathcal{L}_{PMD} = {} & (1 - \boldsymbol{\alpha}_i[\ell]) \times \mathcal{L}_{ce}(B_l; \theta_i) \\ & + \boldsymbol{\alpha}_i[\ell] \;\; \times \mathcal{L}_{kd}(B_\ell; \theta_i, \theta_{3-i}),\end{aligned} \tag{4}$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^{|L|}$ is the multilingual distillation weight vector of θ_i and $\boldsymbol{\alpha}_i[\ell] \in [0, 1]$ is the distillation weight for language pair ℓ. $\boldsymbol{\alpha}_i[\ell]$ is crucial as controlling the extent how much θ_i should learn from θ_{3-i} in direction ℓ. When $\boldsymbol{\alpha}_i[\ell] = 0$, θ_i does not acquire information from θ_{3-i} in ℓ. The values of $\boldsymbol{\alpha}_i$ are determined by the specific strategy. We summarize the whole training framework in Alg.1.

3.2 UNI-PMD and BI-PMD

Multilingual distillation weights $\boldsymbol{\alpha}_i$ play important roles in Pareto-MD. We present two strategies, unidirectional Pareto mutual distillation (UNI-PMD) and bidirectional Pareto mutual distillation (BI-PMD), for determining the values of $\boldsymbol{\alpha}_i$ based on different design philosophies.

UNI-PMD. UNI-PMD is designed based on the intuition that *each model should only learn from the strengths and avoid mimicking the shortcomings of the other model.* Therefore, in each translation direction ℓ, UNI-PMD lets the model that performs less well, denoted by θ_ℓ^{worse}, be distilled by the model that performs better in this direction, denoted by θ_ℓ^{better}, via setting a positive distillation weight. Conversely, UNI-PMD zeros the weight to forbid θ_ℓ^{better} from being influenced by θ_ℓ^{worse}.

Formally, given multilingual validation datasets $\{D_\ell^{valid}\}_{\ell=1}^{|L|}$ and a pre-defined hyper-parameter $\alpha \in [0, 1]$, in each direction $\ell \in L$, UNI-PMD sets the distillation weight of θ_i as:

$$\boldsymbol{\alpha}_i[\ell] = \alpha \times \mathbb{1}\{i = \operatorname*{arg\,max}_{j \in \{1,2\}} \mathcal{L}_{ce}(D_l^{valid}; \theta_j)\}, \tag{5}$$

where the $\mathbb{1}\{\cdot\}$ is an indicator function, indicating whether the model θ_i performs less well on the translation of ℓ. UNI-PMD updates the distillation weights every $\mathcal{T}$ steps.

BI-PMD. Besides, we design another strategy BI-PMD based on the hypothesis that among the two models that are trained with Pareto-MD, in

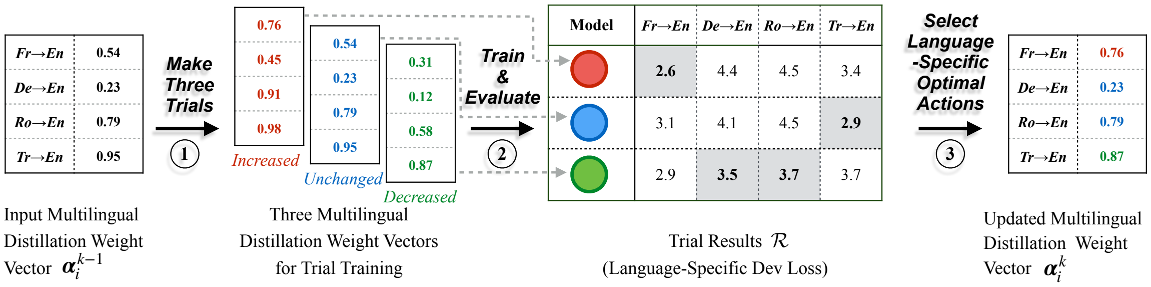

Fr→En
De→En
Ro→En
Tr→En
0.54
0.23
0.79
0.95
Input Multilingual Distillation Weight Vector α_i^{k-1}
Make Three Trials
①
0.76
0.45
0.91
0.98
Increased
0.54
0.23
0.79
0.95
Unchanged
0.31
0.12
0.58
0.87
Decreased
Three Multilingual Distillation Weight Vectors for Trial Training
Train & Evaluate
②
Model
2.6
4.4
4.5
3.4
3.1
4.1
4.5
2.9
2.9
3.5
3.7
3.7
Trial Results $\mathcal{R}$ (Language-Specific Dev Loss)
Select Language -Specific Optimal Actions
③
0.76
0.23
0.79
0.87
Updated Multilingual Distillation Weight Vector α_i^k

Figure 3: Process of AUTO-PMD updating the distillation weights. At the k-th update, AUTO-PMD makes three trials that perform three actions to all language pairs' weights and then train the current model. Finally, the language-specific optimal actions are selected to update the previous weights. Note that the value of each weight will change by different magnitudes when increased or decreased due to the non-linear nature of sigmoid function.

each translation direction ℓ, θ_ℓ^{worse} is also possible to improve θ_ℓ^{better} via knowledge distillation. This hypothesis is motivated by the recently proposed theoretical framework of *Multi-View Data* (Allen-Zhu and Li, 2020; He and Ozay, 2021), which theoretically reveals that each well-trained network only captures a different subset of relevant features, limiting their generalization. The mechanism of knowledge distillation is to help one model to learn the relevant features of another model.

The discovery motivates us to suspect that θ_ℓ^{worse} can also improve θ_ℓ^{better} using distillation, as θ_ℓ^{worse} may possess relevant features that θ_ℓ^{better} lacks. Therefore, BI-PMD allows θ_ℓ^{worse} to affect θ_ℓ^{better} in direction ℓ. Our implementation is simple: BI-PMD sets all distillation weights to a positive value. Formally, given a hyper-parameter α, the distillation weight of θ_i in direction ℓ is:

$$\boldsymbol{\alpha}_i[\ell] = \alpha, \tag{6}$$

meaning that each model affects the other equally.

3.3 AUTO-PMD

Desiderata. Both UNI-PMD and BI-PMD determine the distillation weights of all translation directions based on a pre-defined hyper-parameter α, which dissatisfies the following three expected properties of distillation weights: 1) **Language-Adaptability**: the optimal distillation weights for different language pairs vary. However, the current strategies set a uniform weight for all language pairs, resulting in sub-optimal performance; 2) **Dynamics**: existing research on mutual distillation uses a fixed distillation weight throughout the training process, which fails to adapt to the evolving models; 3) **Generality**: it is empirically discovered that the optimal value of distillation weight varies across different datasets, incurring the extra cost of the manual hyper-parameter search. To satisfy these three properties, we propose Automatic Pareto Mutual Distillation (AUTO-PMD) to automatically decide the value of each direction's distillation weight according to training dynamics.

Approach. AUTO-PMD updates multilingual distillation weight vector $\boldsymbol{\alpha}_i$ every $\mathcal{T}$ steps. We denote the values of $\boldsymbol{\alpha}_i$ after the k-th update by $\boldsymbol{\alpha}^k$. Note that the subscript i of $\boldsymbol{\alpha}_i$ is omitted for clarity. The update process is modeled as Markov Chain (Norris and Norris, 1998). All distillation weights are initialized at the beginning of training as a small value, *i.e.*, $\boldsymbol{\alpha}^0[\ell] = 0.1$. Three actions on distillation weight are defined:

$$\mathcal{F} = \{f_\uparrow(\cdot), f_\downarrow(\cdot), f_=(\cdot)\}, \tag{7}$$

which aim to increase, decrease and keep the value of distillation weight unchanged. At the k-th update, AUTO-PMD decides the values of $\boldsymbol{\alpha}^k$ according to the previous state $\boldsymbol{\alpha}^{k-1}$. We exemplify the process of each update step in Fig. 3 and precisely describe it in Alg. 2. As illustrated in Fig. 3, the update process is divided into three steps.

In the first step, given the previous distillation weights $\boldsymbol{\alpha}^{k-1}$, AUTO-PMD makes three trials, generating three multilingual distillation weight vectors for the trial training of the next step. Each vector is obtained by performing an action (*e.g.*, increasing) on all values of $\boldsymbol{\alpha}^{k-1}$. These three vectors, corresponding to three colorful vectors in Fig. 3, form a set which is referred to as search space $\widetilde{O}^k$. In fact, the trial training of next step should be conducted over the entire search space O^k, which is the Cartesian product of possible subsequent states of each language-specific distillation

Algorithm 2: AUTO-PMD

```
Input      : Multilingual trial datasets $\{D_\ell^{trial}\}_{\ell=1}^{|L|}$,
             validation datasets $\{D_\ell^{valid}\}_{\ell=1}^{|L|}$, the
             training model $\theta_1$and $\theta_2$, search space $\widetilde{O}_1^k$,
             $\widetilde{O}_2^k$, distillation weights $\boldsymbol{\alpha}_1^{k-1}, \boldsymbol{\alpha}_2^{k-1}$
Output     : $\boldsymbol{\alpha}_1^k, \boldsymbol{\alpha}_2^k$
Initialize: Initialize trial results $\mathcal{R} \in \mathbb{R}^{|L| \times |\widetilde{O}_i^k|}$ to a
             zero matrix
for $i \leftarrow 1$ to $2$ do
    for $j \leftarrow 1$ to $|\widetilde{O}_i^k|$ do
        $\boldsymbol{\alpha}_i' \leftarrow \widetilde{O}_i^k[j]$
        Copy model $\theta_i' \leftarrow \theta_i$
        Train $\theta_i'$ on $D^{trial}$ for one epoch using
          teacher model $\theta_{3-i}$ and $\boldsymbol{\alpha}_i'$ with Eq.4
        for $\ell \leftarrow 1$ to $|L|$ do
            $\mathcal{R}[\ell][j] \leftarrow \mathcal{L}_{ce}(D_\ell^{valid}; \theta_i')$
        end
    end
    for $\ell \leftarrow 1$ to $|L|$ do
        $\hat{j} \leftarrow \arg\min_j \mathcal{R}[\ell][j]$
        $\boldsymbol{\alpha}_i^k[\ell] \leftarrow \widetilde{O}_i^k[\hat{j}][\ell]$
    end
end
```

weight $\boldsymbol{\alpha}^{k-1}[\ell]$:

$$O^k = \bigtimes_{\ell \in L} \{f(\boldsymbol{\alpha}^{k-1}[\ell]) \mid f \in \mathcal{F}\}. \tag{8}$$

However, this search space grows exponentially as the number of languages increases, that is, $|O^k| = |\mathcal{F}|^{|L|}$. To overcome the non-trivial cost, the subspace $\widetilde{O}^k$ is adopted. Furthermore, we prove that based on the *Distillation Weights Independence* assumption, the optimal solution searched in $\widetilde{O}^k$ is equivalent to that of O^k. The mathematical description of this assumption and the proof are demonstrated in §A.

Next, AUTO-PMD uses each distillation weight vector in $\widetilde{O}^k$ to train the current model on trial set D^{trial}, which is constructed by sampling ρ of D^{train}, for one epoch. The three trained models are evaluated on the validation set, and the language-specific dev losses of these models form a matrix, which is represented by trial results $\mathcal{R} \in \mathbb{R}^{|\widetilde{O}^k| \times |L|}$. The model training of this step incurs overhead, which is proportional to the value of $\rho \times |\widetilde{O}^k|$. In this work, we set $\rho = 0.1$. Thereby, the extra overhead is 30% of the actual model training.

Finally, the language-specific optimal actions are selected according to the trial results and then performed on $\boldsymbol{\alpha}^{k-1}[\ell]$, obtaining the results of $\boldsymbol{\alpha}^k[\ell]$. We exemplify this step with Fig. 3. The red model, trained using the increased version of $\boldsymbol{\alpha}^{k-1}$ (the vector in red), achieves the best performance of *Fr→En*. Thus, the $\boldsymbol{\alpha}^k[\ell]$ of *Fr→En* is obtained by increasing the $\boldsymbol{\alpha}^{k-1}[\ell]$ of *Fr→En*.

Implementation of Actions. As aforementioned, three actions for updating distillation weights are defined (in Eq.7). The $f_=(\cdot)$ is simple:

$$f_=(\boldsymbol{\alpha}[\ell]) = \boldsymbol{\alpha}[\ell]. \tag{9}$$

For $f_\uparrow(\cdot)$ and $f_\downarrow(\cdot)$, it must ensure that the output is always between $[0, 1]$. Therefore, the input is first mapped into $(-\infty, +\infty)$ using the inverse of sigmoid function and then increased/decreased the value by μ, named step size. Finally, the increased/decreased value is mapped back into $[0, 1]$ using sigmoid function. Formally:

$$f_\uparrow(\boldsymbol{\alpha}[\ell]) = \sigma(\sigma^{-1}(\boldsymbol{\alpha}[\ell]) + \mu) \tag{10}$$

$$f_\downarrow(\boldsymbol{\alpha}[\ell]) = \sigma(\sigma^{-1}(\boldsymbol{\alpha}[\ell]) - \mu) \tag{11}$$

where $\sigma(\cdot)$ is sigmoid function. The step size μ is crucial for weights search. A smaller step size could improve the precision of searched weights while may delay convergence to the optimal weight. Therefore, we design a **step size scheduler**, setting a large step size in the early training stage and then deducing the step size:

$$\mu = \sqrt{\frac{\mathcal{T}_{max} - t}{\mathcal{T}_{max}}} \tag{12}$$

where $\mathcal{T}_{max}$ is the max training steps.

4 Experiments

4.1 Settings

Datasets. We conduct experiments on two datasets: the WMT-6 dataset provided by Huang et al. (2022) and the widely-used TED-8-Diverse dataset constructed by Wang et al. (2020). The WMT-6 dataset involves the language pairs of 3 LRLs (*et*, *ro*, *tr*) and 3 HRLs (*fr*, *de*, *zh*) to English. This dataset has around 5M training sentences from parallel corpora that WMT provides over multiple years, and the corresponding validation and test sets are used. The data statistics are detailed in Appendix B. The TED-8-Diverse contains the language pairs of 4 LRLs (*bos*, *mar*, *hin*, *mkd*) and 4 HRLs (*ell*, *bul*, *fra*, *kor*) to English. This dataset comprises around 570K sentence pairs. The data statistics and the interpretation of language codes are demonstrated in Appendix B. Compared to TED-8-Diverse, the size of WMT-6 dataset is more considerable and distributed more unevenly.

Method	Sampling	WMT-6		TED-8-Diverse	
		Many-to-One	One-to-Many	Many-to-One	One-to-Many
Existing Balancing Training Strategies					
Temperature Sampling	$\tau = 1$	20.57	18.92	29.00	22.75
Temperature Sampling	$\tau > 1$	19.93	18.63	28.35	22.23
MultiDDS-S (Wang et al., 2020)*	*dyn.*	–	–	27.00	18.24
MultiUAT (Wu et al., 2021)*	*dyn.*	–	–	27.83	19.76
CCL-M (Zhang et al., 2021)*	*dyn.*	–	–	28.34	19.53
χ-IBR (Zhou et al., 2021)*	*dyn.*	–	–	29.74	23.44
Existing Knowledge Distillation-based Strategies					
Multi-Distill (Tan et al., 2019)	$\tau = 1$	20.18	18.57	29.52	22.31
LSSD (Huang et al., 2022)	$\tau = 1$	21.17	19.76	30.77	**23.55**
Our Pareto Mutual Distillation					
Uni-PMD	$\tau = 1$	20.76†	18.96	29.76†	22.92
	$\tau > 1$	21.74†	19.76†	29.97†	22.91
Bi-PMD	$\tau = 1$	21.61†	19.53†	30.31†	23.00†
	$\tau > 1$	21.92†	20.09†	30.42†	22.77
Auto-PMD	$\tau = 1$	21.89†	20.16†	**31.05**†	23.31†
	$\tau > 1$	**22.39**†	**20.48**†	30.71†	23.28†

Table 1: BLEU scores on the WMT-6 and TED-8-Diverse dataset. Bold indicates the highest BLEU score in each setting. '*' means results taken from the original paper. '†' indicates significantly better than temperature-based sampling with t-test $p < 0.001$. The temperature-based sampling is tried with $\tau = \{1, 5\}$ on WMT-6 and $\tau = \{1, 3\}$ on TED-8-Diverse. For each of our approaches, the first row is the result of model-1, and the second row is the result of model-2. '*dyn.*' is the abbreviation for "dynamic sampling."

For each dataset, our approach is evaluated in two multilingual translation scenarios: 1) Many-to-One (M2O): translating multiple languages to English in this work; 2) One-to-Many (O2M): translating English to other languages.

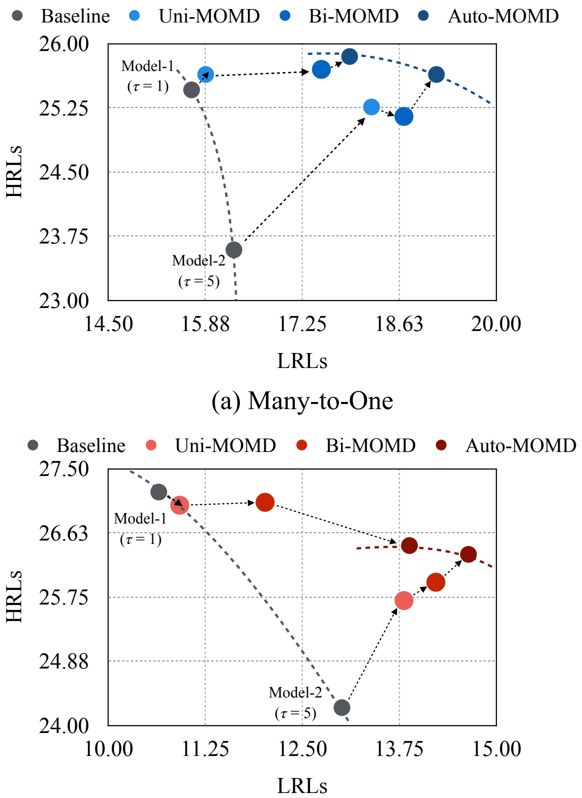
Hyper-parameters. Even though our proposed training framework can be applied to any model architecture, we verify its effectiveness on the popular Transformer (Vaswani et al., 2017) implemented in fairseq (Ott et al., 2019) with the base version. We use the same model configuration, hyper-parameters, and preprocess procedure as those of Huang et al. (2022) for all baselines and our method. The only difference is that the dropout rate is modified into 0.2 on WMT-6, to accelerate the convergence without performance loss. The complete set of hyper-parameters is demonstrated in Appendix C. The performance is evaluated with the BLEU score (Papineni et al., 2002) using the SacreBLEU toolkit (Post, 2018).

As illustrated in §3.1, our Pareto-MD trains two models using different sampling distributions, P_1 and P_2, and we adopt temperature-based sampling with different values of τ to produce these two distributions. We set $\tau = 1$ for P_1 and $\tau = 5$ for P_2 on WMT-6. On TED-8-Diverse, we set $\tau = 1$ for model-1 and $\tau = 3$ for model-2 since an overly large value leads to poor performance. For the Uni-PMD and Bi-PMD, we manually search the optimal α (in Eq.5 and Eq.6) among $\{0.2, 0.4, 0.6, 0.8\}$. The update interval of distillation weights $\mathcal{T}$ is set to the step number of one epoch.

Baselines. We primarily compare our Pareto-MD with: (1) Temperature-based Sampling: the method most related to our work; 2) χ-IBR (Zhou et al., 2021), the state-of-the-art (SOTA) dynamic sampling method, which enables the balancing training based on *distributionally robust optimization*; 3) LSSD (Huang et al., 2022), another distillation-based training strategy which achieves SOTA performance on TED-8-Diverse and WMT-6 dataset via alleviating the convergence inconsistency problem of MNMT using self-distillation. More details of baselines are demonstrated in Appendix D.

4.2 Main Results

We summarize the main results in Table 1. As we observed, our methods significantly outperform the temperature-based sampling under M2O and O2M settings on both datasets. The model-2 trained with Auto-PMD has improved by up to +2.46 BLEU under the M2O setting of WMT-6.



(a) Many-to-One

(b) One-to-Many

Figure 4: Multilingual performance Pareto frontier on the WMT-6 dataset. Gray dotted curves indicate the Pareto frontier of baselines and the colorful ones mark the frontier made by AUTO-PMD. This figure shows that the Pareto frontier is pushed outwards significantly.

Method	Sampling	BLEU
Vanilla MD	$\tau = 1$ $\tau = 1$	20.93 20.97
Vanilla MD	$\tau = 5$ $\tau = 5$	21.13 21.29
BI-PMD	$\tau = 1$ $\tau = 5$	21.61 21.92
AUTO-PMD	$\tau = 1$ $\tau = 5$	21.89 **22.39**

Table 2: Comparison between our method with vanilla mutual distillation (Vanilla MD) under the Many-to-One setting of the WMT-6 dataset.

Furthermore, Pareto-MD achieves higher BLEU scores than previous methods in most settings. At best, AUTO-PMD outperforms the previous SOTA (LSSD) by +1.22 BLEU scores under the M2O setting of WMT-6. When comparing UNI-PMD and BI-PMD, it is obvious that BI-PMD consistently exceeds UNI-PMD, verifying the motivation that the worse model is also possible to improve the better model via knowledge distillation. AUTO-PMD further surpasses BI-PMD by +0.3∼0.5 BLEU. This improvement proves that our automatic search of distillation weights is indeed reliable. Moreover, AUTO-PMD is more general than UNI-PMD and BI-PMD since it eliminates the need to search for the hyper-parameter α manually[3].

5 Analysis

5.1 Visualization of Pareto Frontier

In order to clearly assess the impact of our methods on HRLs and LRLs, we visualize the Pareto frontier in Fig. 4. Three important observations can be drawn: 1) overall, the model-1 has been significantly shifted right, and the model-2 has been shifted upwards, proving that Pareto-MD effectively alleviates the shortcomings of each model as we expected; 2) both of model-1 and model-2 are shifted right beyond the original model-2, indicating that the performance of LRLs is improved beyond the original performance bound. The reason may be that the transfer learning from HRLs to LRLs is more effective when the model achieves high performance on both HRLs and LRLs; 3) the model-1 degenerates on the translation of HRLs in the O2M setting. One potential cause is the representation space of HRLs undergoing more intense squeezing in the O2M than in the M2O when the model learns well on LRLs.

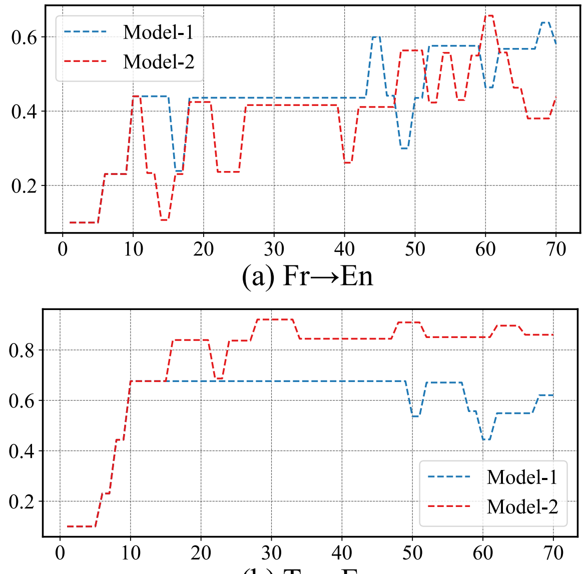
5.2 Effect of Diverse Sampling Strategies

In the Pareto-MD training framework, two models corresponding to different Pareto optimal solutions are trained collaboratively using distinct training distributions. One natural question that arises is, how would the performance be affected if we trained two models with the same training distribution? This approach, in fact, degenerates into the vanilla mutual distillation method. Therefore, we conduct a comparison experiment on the WMT-6 dataset (M2O setting) shown in Table 2. The results indicate that vanilla mutual distillation underperforms our BI-PMD by about 0.6 BLEU, which supports the effectiveness of using different sampling distributions for our Pareto-MD. Moreover, we propose AUTO-PMD to improve vanilla mutual distillation by +1.1 BLEU totally.

5.3 Evolution of Distillation Weights

To better understand the process of AUTO-PMD, we visualize the automatically searched distillation weights in Fig. 5. As it depicts, the distillation

[3] The effect of α is shown in Appendix F.



(a) Fr→En

(b) Tr→En

Figure 5: Visualization of automatically search distillation weights in the many-to-one setting of WMT-6 dataset. Due to the space limitation, we only show the weights of one HRL (Fr→En) and one LRL (Tr→En)

weights constantly vary to adapt the dynamic models with a decreasing variance made by the decay of search step size (Eq.12). Besides, it is discovered that the low-resource Tr→En translation favors a higher value of distillation weight than the high-resource Fr→En translation. This phenomenon makes sense since LRLs suffer from more serious over-fitting (Huang et al., 2022), requiring stronger distillation learning.

5.4 Effect of Step Size Scheduler μ

The performance of different step size schedulers is listed in Table 3. The simple scheduler-1 fixes the step size to 1.0, performing relatively poorly. The scheduler-2 decreases the step size from 1.0 to 0.2. The scheduler-4 decreases the step size from 1.0 to 0.0, achieving the best performance. The scheduler-3 also decrease the step size from 1.0 to 0.0, while not performing searching of distillation weights at the end of training. We finally adopt the scheduler-4 in our AUTO-PMD.

#	Scheduler	BLEU ($\tau = 1$ / $\tau = 5$)
1	$\mu = 1$	20.71 / 21.80
2	$\mu = \sqrt{\frac{\mathcal{T}_{max}-0.8\times t}{\mathcal{T}_{max}}}$	21.90 / 22.21
3	$\mu = max(\sqrt{\frac{\mathcal{T}_{max}-1.2\times t}{\mathcal{T}_{max}}}, 0)$	21.74 / 22.31
4	$\mu = \sqrt{\frac{\mathcal{T}_{max}-t}{\mathcal{T}_{max}}}$	21.89 / **22.39**

Table 3: Effect of step size scheduler μ in the many-to-one translation of WMT-6 dataset. We have tried for four implementations of the step size scheduler.

6 Related Work

For a long time, data imbalance has been a problem hindering multilingual models from performing evenly across different languages. Existing methods pursue balanced performance via designing heuristics (Arivazhagan et al., 2019) or automatic sampling strategies (Arivazhagan et al., 2019; Wang et al., 2020; Zhou et al., 2021; Wu et al., 2021; Zhang et al., 2021). For example, Wang et al. (2020) design a Reinforce Learning based method to automatically adjust the sampling probability of each language pair towards an overall optimal solution. Zhou et al. (2021) vary the distribution via distributional robust optimization. However, their improvement is limited since increasing the training weights of some languages leads to relative decreases in the weights of other languages, resulting in a trade-off on the Pareto frontier. Different from their methods, we overcome this issue by training two models collaboratively.

Before our work, there were two approaches also based on knowledge distillation in MNMT. Tan et al. (2019) use pre-defined bilingual models to teach the multilingual model via knowledge distillation. Huang et al. (2022) propose language-specific self-distillation to remedy the convergence inconsistency problem in MNMT using self-distillation. Our Pareto-MD is an extension of mutual distillation on the Pareto optimization problems.

7 Conclusion

In this work, we propose a training framework Pareto-MD to reach a higher Pareto frontier for MNMT. The core of Pareto-MD is the synergy between diverse Pareto optimal solutions via mutual distillation. Besides, we design a novel strategy for deducing distillation weights automatically, achieving better performance and getting rid of hyper-parameter searching. Experimental results on the WMT and TED datasets show the effectiveness of our method. Even though we experiment with training two models in this work, our method can naturally apply to train more models. In the future, we are interested in exploring how to apply our Pareto-MD to the training of large language models (Zhao et al., 2023).

Limitations

Our Pareto-MD doubles computational cost due to training two models simultaneously, which can be a limitation of our approach. However, Pareto-MD obtains significant improvement that is hard to achieve for previous methods of training individual models, thus worthy. Besides, our approach would not necessarily result in double training time because these two models can be trained in parallel as implemented by Guo et al. (2020). Moreover, Pareto-MD does not affect inference efficiency.

Acknowledgements

Xiaocheng Feng is the corresponding author of this work. We thank the anonymous reviewers for their insightful comments. This work was supported by the National Key R&D Program of China via grant 2020AAA0106502, National Natural Science Foundation of China (NSFC) via grant 62276078, the Key R&D Program of Heilongjiang via grant 2022ZX01A32 and the International Cooperation Project of PCL, PCL2022D01.

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.

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. 2020. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Conference on Spoken Language Translation*, Seattle, Washington D.C. International Workshop on Spoken Language Translation.

Bobby He and Mete Ozay. 2021. Feature kernel distillation. In *International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

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*. Association for Computational Linguistics.

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.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical*

Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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.

James R Norris and James Robert Norris. 1998. *Markov chains*. 2. Cambridge university press.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. 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.

Minghao Wu, Yitong Li, Meng Zhang, Liangyou Li, Gholamreza Haffari, and Qun Liu. 2021. Uncertainty-aware balancing for multilingual and multi-domain neural machine translation training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7291–7305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from Microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 446–455, Online. Association for Computational Linguistics.

Mingliang Zhang, Fandong Meng, Yunhai Tong, and Jie Zhou. 2021. Competence-based curriculum learning for multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2481–2493, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

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.

A Equivalence Between Searching in O^k and $\widetilde{O}^k$

As illustrated in §3.3, our strategy AUTO-PMD first searches the language-specific optimal multilingual distillation weight vector $\hat{\boldsymbol{\alpha}}^{\ell}$ for each translation direction ℓ from a search space and then take the $\hat{\boldsymbol{\alpha}}^{\ell}[\ell]$ as the searching result of $\boldsymbol{\alpha}^k[\ell]$. To search the optimal solution, the search space should be the entire space O^k, which is formalized as:

$$O^k = \bigtimes_{\ell \in L} \{f(\boldsymbol{\alpha}^{k-1}[\ell]) \mid f \in \mathcal{F}\},$$

However, the size of O^k grows exponentially as the number of languages increases. Therefore, we instead search in $\widetilde{O}^k$, a subset of O^k, which is formalized as:

$$\begin{aligned}\widetilde{O}^k = \{\ &\{\ f_{\uparrow}(\boldsymbol{\alpha}^{k-1}[\ell])\ \}_{\ell \in L},\\ &\{\ f_{\downarrow}(\boldsymbol{\alpha}^{k-1}[\ell])\ \}_{\ell \in L},\\ &\{\ f_{=}(\boldsymbol{\alpha}^{k-1}[\ell])\ \}_{\ell \in L}\ \}.\end{aligned}$$

In this section, we initially give a formal definition of the searching process. Subsequently, the Distillation Weights Independence (DWI) assumption is introduced. Ultimately, we prove the equivalence between searching in O^k and $\widetilde{O}^k$ based on the DWI assumption.

Definition A.1 (Searching Process)**.** Given the multilingual trial set $D^{trial} = \{D^{trial}_{\ell}\}_{\ell=1}^{|L|}$, validation set $D^{valid} = \{D^{valid}_{\ell}\}_{\ell=1}^{|L|}$, student mode θ^S, teacher model θ^T, and the search space O , for each translation direction ℓ, the searching process of $\boldsymbol{\alpha}^k[\ell]$ is:

$$\begin{aligned}\boldsymbol{\alpha}^k[\ell] &= \hat{\boldsymbol{\alpha}}^{\ell}[\ell]\\ \hat{\boldsymbol{\alpha}}^{\ell} &= \arg\min_{\boldsymbol{\alpha} \in O} \mathcal{L}_{ce}(D^{valid}_{\ell}; \hat{\theta}(\boldsymbol{\alpha}))\\ \hat{\theta}(\boldsymbol{\alpha}) &= \arg\min_{\theta} \mathcal{L}_{PMD}(D^{trial}; \theta^S, \theta^T, \boldsymbol{\alpha}).\end{aligned}$$

Hypothesis A.1 (Distillation Weights Independence)**.** *Given two multilingual distillation weight vectors $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$:*

$$\begin{aligned}&\exists \ell \in L, \boldsymbol{\alpha}_1[\ell] = \boldsymbol{\alpha}_2[\ell]\\ \Rightarrow\ &\mathcal{L}_{ce}(D^{valid}_{\ell}; \hat{\theta}(\boldsymbol{\alpha}_1)) = \mathcal{L}_{ce}(D^{valid}_{\ell}; \hat{\theta}(\boldsymbol{\alpha}_2))\end{aligned}$$

Theorem A.1. *Let $\hat{\boldsymbol{\alpha}}^{\ell}[\ell]$ denote the searching result in the search space O^k for direction ℓ, $\tilde{\boldsymbol{\alpha}}^{\ell}[\ell]$ denotes the searching result in the search space $\widetilde{O}^k$ for direction ℓ, based on the Distillation Weights Independence assumption, it is satisfied that:*

$$\hat{\boldsymbol{\alpha}}^{\ell}[\ell] = \tilde{\boldsymbol{\alpha}}^{\ell}[\ell].$$

Proof. Let $\hat{\boldsymbol{\alpha}}^{\ell}[\ell] = \hat{f}^l(\boldsymbol{\alpha}^{k-1}[\ell])$, where $\hat{f}^l \in \mathcal{F}$ is the language-specific action, the following equation holds:

$$\mathcal{L}_{ce}(D^{valid}_{\ell}; \theta(\hat{\boldsymbol{\alpha}}^{\ell})) = \mathcal{L}_{ce}(D^{valid}_{\ell}; \theta(\{\hat{f}^l(\boldsymbol{\alpha}^{k-1}[\ell'])\}_{\ell' \in L})),$$

based on hypothesis A.1. Because $\{\hat{f}^l(\boldsymbol{\alpha}^{k-1}[\ell'])\}_{\ell' \in L} \in \widetilde{O}^k$, and $\widetilde{O}^k \subseteq O^k$, then we can infer that:

$$\begin{aligned}\Rightarrow\quad \mathcal{L}_{ce}(D^{valid}_{\ell}; \{\hat{f}^l(\boldsymbol{\alpha}^{k-1}[\ell'])\}_{\ell' \in L}) &= \min_{\boldsymbol{\alpha} \in \widetilde{O}^k} \mathcal{L}_{ce}(D^{valid}_{\ell}; \hat{\theta}(\boldsymbol{\alpha}))\\ \Rightarrow\quad \{\hat{f}^l(\boldsymbol{\alpha}^{k-1}[\ell'])\}_{\ell' \in L} &= \arg\min_{\boldsymbol{\alpha} \in \widetilde{O}^k} \mathcal{L}_{ce}(D^{valid}_{\ell}; \hat{\theta}(\boldsymbol{\alpha}))\\ \Rightarrow\quad \hat{f}^l(\boldsymbol{\alpha}^{k-1}[\ell]) &= \tilde{\boldsymbol{\alpha}}^{\ell}[\ell]\\ \Rightarrow\quad \hat{\boldsymbol{\alpha}}^{\ell}[\ell] &= \tilde{\boldsymbol{\alpha}}^{\ell}[\ell]\end{aligned}$$

□

B Data Statistics

We list data statistic of TED-8-Diverse dataset in Table 4. Data statistics of WMT-6 dataset is listed in Table 5.

Language	Num
bos (Bosnian)	5,664
mar (Marathi)	9,840
hin (Hindi)	18,798
mkd (Macedonian)	25,335
ell (Greek)	134,327
bul (Bulgarian)	174,444
fra (French)	192,304
kor (Korean)	205,640

Table 4: Data statistics for the TED-8-Diverse dataset. 'num' refers to the number of sentence pairs in the training set.

Language	Data Source	Num
tr (Turkish)	WMT17	5,000
ro (Romanian)	WMT16	10,000
et (Estonian)	WMT18	80,000
zh (Chinese)	WMT17	400,000
de (German)	WMT14	1,500,000
fr (French)	WMT14	3,000,000

Table 5: Data statistics for the WMT dataset. 'num' refers to the number of sentence pairs in the training set.

C Hyper-parameters

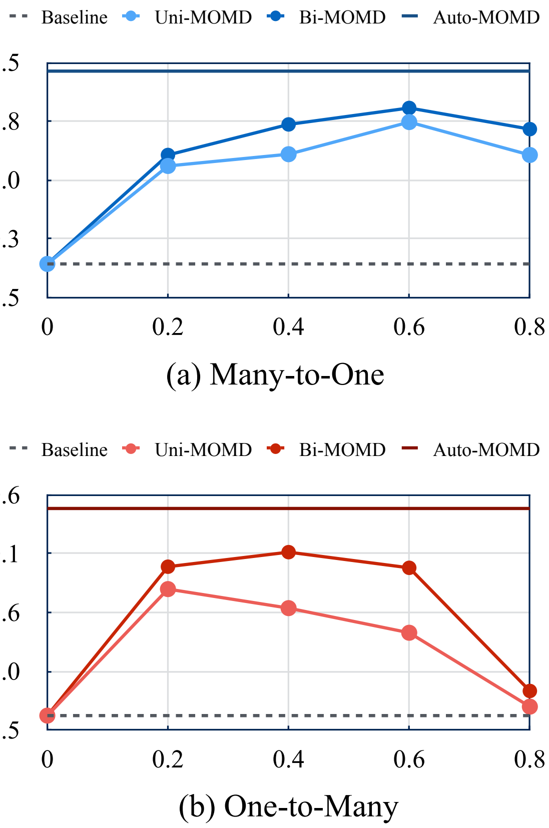
In this section, we report the hyper-parameters used in our experiments.

- We adopt the base-version of Transformer architecture with 6 layers encoders/decoders and 8 attention heads.

- The embedding dimension is 512 and the Feed-Forward Network has a dimension of 2048.

- We train models with learning rate $\eta = 0.0015$ and use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.98$, and the same learning rate schedule as Vaswani et al. (2017).

- Batch size is set to 64K and half-precision training is adopted (Ott et al., 2018).



(a) Many-to-One

(b) One-to-Many

Figure 6: Effect of different values of α on WMT-6 dataset. For clarity, we only depict the results of model-2 trained with $\tau = 5$.

- For regularization, we use the label smoothing as 0.1 (Szegedy et al., 2016). We set the dropout as 0.3 (Srivastava et al., 2014) on the TED-8-Diverse dataset and as 0.2 on the WMT-6 dataset.

- Models are trained for 70 epochs on WMT-6 and 300 epochs on TED-8-Diverse according to the convergence.

- For TED-8-Diverse, we preprocess sentececes using sentencepiece (Kudo and Richardson, 2018) with a vocabulary size of $8K$ for each language. For WMT-6, the vocabulary size is $64K$ for all languages.

- For inference, we use beam search with beam size 5.

All models are trained on Tesla V100 GPUs.

D Details about Baselines

For temperature-based sampling (Arivazhagan et al., 2019), we adopt the official implementation in fairseq. LSSD is re-implemented successfully with the code released by Huang et al. (2022).

Setting	Method	Sampling	fr	de	zh	et	ro	tr	*Avg.*
M2O	*Temperature Sampling*	$\tau = 1$	34.40	28.70	13.27	16.41	22.65	7.99	20.57
		$\tau > 1$	31.59	26.61	12.56	16.48	23.06	9.29	19.93
	AUTO-PMD	$\tau = 1$	**34.96**	**28.79**	13.81	17.9	25.22	10.65	21.89
		$\tau > 1$	34.09	28.77	**14.05**	**19.22**	**26.62**	**11.60**	**22.39**
O2M	*Temperature Sampling*	$\tau = 1$	**36.16**	**23.89**	21.49	11.53	14.85	5.58	18.92
		$\tau > 1$	31.21	20.76	20.76	13.28	17.54	8.20	18.63
	AUTO-PMD	$\tau = 1$	35.38	23.12	20.84	13.2	18.79	9.65	20.16
		$\tau > 1$	34.47	23.00	**21.51**	**14.15**	**19.54**	**10.23**	**20.48**

Table 6: BLEU score per language pair on the WMT-6 dataset. '*Avg.*' is the abbreviation of "average values". Bold indicates the best performance of each language pair. Languages are sorted in decreasing order from left to right according to data size.

Setting	Method	Sampling	kor	fra	bul	ell	mkd	hin	mar	bos	*Avg.*
M2O	*Temperature Sampling*	$\tau = 1$	19.73	40.73	39.74	38.71	34.34	23.38	11.13	24.88	29.08
		$\tau > 1$	18.79	40.1	39.00	38.11	32.89	22.55	10.36	24.98	28.35
	AUTO-PMD	$\tau = 1$	**21.14**	**42.41**	**41.52**	**40.67**	**36.49**	**25.9**	12.32	27.94	**31.05**
		$\tau > 1$	20.51	42.03	40.93	40.00	36.04	25.71	**12.44**	**28.02**	30.71
O2M	*Temperature Sampling*	$\tau = 1$	9.06	40.26	36.10	33.63	25.67	15.56	4.90	16.82	22.75
		$\tau > 1$	8.87	39.96	35.91	33.31	24.35	14.81	4.75	15.87	22.23
	AUTO-PMD	$\tau = 1$	**9.13**	**40.94**	**36.56**	**34.03**	27.15	15.89	**5.13**	17.64	**23.31**
		$\tau > 1$	8.90	40.65	36.55	33.64	**27.44**	**16.29**	4.90	**17.89**	23.28

Table 7: BLEU score per language pair on the TED-8-DIVERSE dataset. '*Avg.*' is the abbreviation of "average values". Bold indicates the best performance of each language pair. Languages are sorted in decreasing order from left to right according to data size.

We have tried to set Dropout rate to $\{0.2, 0.3\}$ for LSSD, and report the best results in terms of BLEU for fair comparison. The code of χ-IBR (Zhou et al., 2021) is also released. However, the result of χ-IBR evaluated in our experiments is lower than the original paper. Therefore, we report the results in the original paper.

E BLEU scores on Individual Languages

In this section, we report the BLEU scores of individual language pairs. For clarity, we only show the results of the temperature-based sampling and our AUTO-PMD. As illustrated in Table. 6 and Table. 7, our method achieves consistent improvements in 3 out of 4 settings.

In the one-to-many setting of WMT-6 dataset, the performance of HRLs (i.e., *fr* and *de*) drops about 0.7 BLEU. This may be due to the parameter interference from the significantly improved LRLs.

F Effect of α for UNI-PMD and BI-PMD

In this section, we show the experimental results of UNI-PMD and BI-PMD with different values of α in Fig. 6. As demonstrated, the value of α is crucial for the performance. The optimal value of α varies across different settings. This conclusion is consistent with former work related to knowledge distillation (Huang et al., 2022), which highlights the importance of deducing distillation weights automatically.

G Other Variants of Mutual Distillation

In this work, we design another two mutual distillation-based strategies beyond AUTO-PMD: Dynamic Mutual Distillation (DYNAMIC-MD) and Language-Specific Mutual Distillation (LSMD). DYNAMIC-MD adopts the same update process of distillation weights as AUTO-PMD. That is, DYNAMIC-MD also makes three trials and uses the

Method	Sampling	BLEU	
		M2O	O2M
AUTO-PMD	$\tau = 1$	21.89	20.16
	$\tau = 5$	**22.39**	**20.48**
DYNAMIC-MD	$\tau = 1$	22.06	20.33
	$\tau = 5$	22.11	20.24
LSMD	$\tau = 1$	21.47	18.94
	$\tau = 5$	21.03	19.46

Table 8: Other variants of mutual distillation designed by us. DYNAMIC-MD is the abbreviation of Dynamic Mutual Distillation. LSMD is the abbreviation of Language-Specific Mutual Distillation.

optimal action to uptate the distillation weight. Differently, DYNAMIC-MD selects a uniform optimal action instead of language-specific optimal actions. LSMD sets fixed and language-specific distillation weights for each language pair. To obtain suitable language-specific distillation weights, we use the distillation weights searched by AUTO-PMD at the last update. The results of these two strategies are listed in Table 8. As the results show, AUTO-PMD achieves higher performance upper-bound than these two strategies.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☒ A2. Did you discuss any potential risks of your work?
We haven't find any risk of our work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
4

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use widely-used and open datasets.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
8

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.