OMNIKIGHT: Multilingual Neural Machine Translation with Language-Specific Self-Distillation

Yichong Huang  Xiaocheng Feng  Xinwei Geng  Bing Qin
Harbin Institute of Technology, China
{ychuang, xcfeng, xwgeng, bqin}@ir.hit.edu.cn

Abstract

Although all-in-one-model multilingual neural machine translation (MNMT) has achieved remarkable progress in the recent years, its selected best overall checkpoint fails to achieve the best performance simultaneously in all language pairs. It is because that the best checkpoints for each individual language pair (i.e., language-specific best checkpoints) scatter in different epochs. In this paper, we present a novel training strategy dubbed Language-Specific Self-Distillation (LSSD) for bridging the gap between language-specific best checkpoints and the overall best checkpoint. In detail, we regard each language-specific best checkpoint as a teacher to distill the overall best checkpoint. Moreover, we systematically explore three variants of our LSSD, which perform distillation statically, selectively, and adaptively. Experimental results on two widely-used benchmarks show that LSSD obtains consistent improvements towards all language pairs and achieves the state-of-the-art. ¹

1 Introduction

Multilingual neural machine translation (MNMT) is a challenging task, which aims to handle translation between more than one language pair with an unified model (Dong et al., 2015; Johnson et al., 2017; Arivazhagan et al., 2019; Fan et al., 2020). It has two main advantages. One is that MNMT achieves promising performance for low-resource languages via transfer learning (Zoph and Knight, 2016; Dabre et al., 2020). And the other one is reducing training and deployment footprints.

Currently, MNMT methods jointly train the model on different language pairs and pick the overall best checkpoint with the best average performance across all language pairs (Johnson et al., 2017; Wang et al., 2020a). However, our pilot study reveals that the overall best checkpoint is not the best performance for each individual language pair, which has its own best checkpoint (i.e., language-specific best checkpoints), and they appear in different training epochs. For example, Figure 1 shows the loss curves of a MNMT model, which is trained to translate 8 languages into English. Obviously, Bosnian (i.e., bos) achieves its best performance in epoch 14 (Upper Left Box), while the overall best checkpoint appears in epoch 24 (Bottom Box). We can see that the loss in epoch 24 for Bosnian is much larger than in epoch 14, which leads to a performance gap between the Bosnian best checkpoint and the overall best checkpoint. Similarly, this gap exists for other language pairs.

In this paper, we propose a novel training strategy Language Specific Self-Distillation (LSSD), which selects these language-specific best checkpoints as teacher models to guide the student model (the overall best checkpoint) separately in each

¹Our code will be released upon acceptance.
individual language pair via self-distillation. In our work, different language teacher models are copies of different training epochs of the student model, where the epoch is determined when the student model obtains the minimum loss value on the corresponding language. And the teacher model is replaced when the student model outperforms its teacher in a language. Besides, as the teacher model may also err in some translation instances. To avoid the negative transferring in these instances, we devise three different instance-selection modes for LSSD: 1) LSSD-WHOLE, which statically performs distillation on the full dataset; 2) LSSD-SELECTIVE, which performs distillation only on data samples where the teacher performs better than its student; 3) LSSD-ADAPTIVE, which performs distillation on the full dataset while with adaptive distillation weights determined by the performance difference between the teacher and student.

We conduct experiments on two widely-used MNMT benchmarks (TED-8-Related and TED-8-Diverse) in two multilingual translation scenarios, i.e., MANY-TO-ONE and ONE-TO-MANY. Experimental results show that, through closing the performance gap between the overall best checkpoint and language-specific best checkpoints (average drop of 46%), LSSD obtains consistent improvement towards all language pairs and achieves state-of-the-art performance.

2 Preliminaries

2.1 Neural Machine Translation

Bilingual neural machine translation translates a source language sentence $x$ into the target language sentence $y$. Given a parallel sentence pairs corpus $D = \{(x, y) \in \mathcal{X} \times \mathcal{Y}\}$, the neural machine translation model is learned with the Maximum Likelihood Estimation (MLE):

$$\theta^* = \arg \max_{\theta} \mathbb{E}_{(x,y) \sim D} \sum_{i \leq |y|} \log P(y_i | x, y_{<i} ; \theta)$$

(1)

where $P(\cdot | \cdot)$ is the conditional probability with model $\theta$, which is usually implemented by the encoder-decoder architecture (Bahdanau et al., 2015; Vaswani et al., 2017).

2.2 Multilingual Neural Machine Translation

Even though conventional NMT has been on par with human beings in single language pair translations, it is unaffordable to deploy one model for each language pair (Xie et al., 2021). MNMT is proposed for alleviating the above problem and has attracted much attention in recent years. In this work, we follow Johnson et al. (2017) to train a MNMT model jointly in training datasets of $n$ language pairs $D_{\text{train}} = D_{\text{train}}^1, ..., D_{\text{train}}^n$, where $D_{\text{train}}^l$ is the dataset of language pair $(S_l, T_l)$. And all parameters are shared for each language pair. To encode and decode multiple languages, a large vocabulary $\mathcal{V}$ is built with diverse languages. And a language tag is appended to the beginning of source sentences to specify the target language. For example, the tag ‘2en’ means translating any source language sentence to the English sentence. Similarly, the MNMT model is also trained with the same objective as Eq.1.

Model Selection Strategy. During training, MNMT models save checkpoints in each epoch and use one dev set (i.e., development set) $D_{\text{dev}} = D_{\text{dev}}^1, ..., D_{\text{dev}}^n$ to evaluate the overall performance of checkpoints for all language pairs. The checkpoint with minimal dev loss is selected as the best checkpoint. The dev loss could be formalized as:

$$L_{\text{dev}}^l(\theta, D_{\text{dev}}^l) = \frac{1}{n} \sum_{l=1}^{n} L(D_{\text{dev}}^l; \theta)$$

(2)

In our work, we consider each language pair with individual dev loss $L_{\text{dev}}^l$, and select the best checkpoint based on this language-specific $L_{\text{dev}}^l$ as the corresponding language teacher.

2.3 Self-Distillation

Self-distillation is a variant of knowledge distillation (Hinton et al., 2015), which is an effective model compression technology that distills knowledge from a high-capacity teacher model into the compact student model. A major difference from vanilla knowledge distillation is that self-distillation uses the same network for both the teacher model and the student model (Zhang et al., 2019; Yang et al., 2019). Yang et al. (2019) use models in earlier epochs to guide the training of later epochs, which boosts the predictive accuracy in image classification by a large margin. In this paper, we extend this idea to make the student model learn multiple different targets at the same training epoch and the number of targets is constantly changing in different epochs.
3 Language-Specific Self-Distillation (LSSD)

In this section, we first demonstrate the overall process of LSSD. Then, we provide detailed formalization of LSSD, with special emphasis on the three different instance-selection modes.

3.1 Overall

In this subsection, we present an overview of our training strategy for MNMT, as illustrated in Figure 2. Specifically, we first take bilingual self-distillation as an example to show the distillation learning process. Then, we describe the multilingual self-distillation model, how it performs self-distillation towards multiple language pairs.

Bilingual LSSD  Traditional bilingual neural machine translation builds a Sequence-to-Sequence model for learning throughout the training process. In our bilingual LSSD, we maintain an additional teacher model and a distillation switch. The teacher model is used to teach the original machine translation model and the switch is designed to decide whether the teacher model is working or not in the current training epoch. We illustrate the bilingual LSSD process in the upper half of Figure 2. The blue box means the student model (MT Model) and the value of loss represents the loss on the dev set of the student in this training epoch. The orange box is the teacher model, and its loss means its dev loss in the corresponding training epoch. The distillation switch is drawn as an ellipse, grey for off and green for on.

In the first $k$ epochs of bilingual LSSD, we do not perform distillation, but iteratively replace the teacher with the student which has a lower loss value. We call this process as “Initial Training Stage”. In each training epoch after initial $k$ epochs, we use the switch to decide whether to perform distillation. The state of the switch is changed by comparing the dev loss of the teacher and the student model in the current training epoch. As we all know, the lower loss, the better. Therefore, if the teacher’s loss is lower than the student’s loss, we turn on the switch, as shown in the $k + 1$ epoch. In fact, we do not perform distillation due to the switch is closed at the beginning of this epoch. In the $k + 2$ epoch, the switch display is turned on, the teacher model distills the student model. And since the teacher’s loss is still lower than the student’s loss, the switch remains turn on. In the $k + 3$ epoch, the switch also shows that it is turned on, the distillation learning is performed. But the teacher’s loss is higher than the student’s loss, we turn off the switch. At the same time, we replace the teacher model with the current student model to complete the teacher updating.

Multilingual LSSD  Multilingual LSSD is a complex version of bilingual LSSD, which needs to maintain multiple language-specific teachers and conduct multi-object distillation learning. We illustrate the multilingual LSSD process in the bottom half of Figure 2. The blue box also means the stu-
We also summarize this process as Algorithm 1.

**Algorithm 1 Language-Specific Self-Distillation algorithm**

**Input**: Training sets \( \{ \mathcal{D}_{\text{train}}^{L_t}\}_{t=1}^T \) and dev sets \( \{ \mathcal{D}_{\text{dev}}^{L_t}\}_{t=1}^T \) for \( L \) language pairs, max training epochs \( K \), training steps of one epoch \( T \), learning rate \( \eta \).

**Initialize**: Randomly initialize MNMT model \( \theta \). For each language pair \( l \in [1, L] \), set language-specific best checkpoint \( \hat{\theta}_l = \theta \), index \( k_l = 1 \), dev loss \( \ell_{l}^{\text{dev}} = +\infty \).

1: for \( k \in [1, K] \) do  
   2: for \( t \in [1, T] \) do  
      3: Randomly sample a language pair \( l \).
      4: Sample a mini-batch of sentence pairs \( B_l^k \).
      5: if \( k - k_l > 1 \) then
         6:     \( \mathcal{L} = \mathcal{L}_{\text{NMT}}(B_l^k; \theta) + \alpha \mathcal{L}_{\text{LSSD}}(B_l^k; \theta, \hat{\theta}_l) \)
      7: else
         8:     \( \mathcal{L} = \mathcal{L}_{\text{NMT}}(B_l^k; \theta) \)
      9: end if
     10: Update \( \theta \): \( \theta = \theta - \eta \cdot \nabla \theta \mathcal{L} \)
   11: end for
   12: for \( l \in [1, L] \) do  
      13: \( \ell_{l}^{\text{dev}} = \mathcal{L}_{\text{NMT}}(D_l^{\text{dev}}; \theta) \)
      14: if \( \ell_{l}^{\text{dev}} < \ell_{l}^{\text{dev}} \) then
         15: \( k_l = k, \hat{\theta}_l = \theta, \ell_{l}^{\text{dev}} = \ell_{l}^{\text{dev}} \)
      16: end if
   17: end for
18: end for

In the first \( k \) epochs, multilingual LSSD also has an “Initial Training Stage”, at which time all teacher models don’t work. After initial \( k \) epochs, the LSSD of each language is working, the student model can be guided by multiple teachers at the same epoch. The different kinds of orange lines mean the language-specific teacher is distilling the student model and the number of working teachers is determined by the language-specific switch, which is consistent with the distillation switch of bilingual LSSD. The blue line represents the language-specific teacher is replaced by the student from the previous training epoch. In this paper, multilingual LSSD is the superposition of multiple bilingual LSSDs and they do not affect each other. We also summarize this process as Algorithm 1.

**3.2 Formalization of LSSD**

Formally, we denote the current model by \( \theta \) and maintain a set of language-specific best checkpoints \( \{ \hat{\theta}_l \}_{l=1}^L \). In the validation stage, i.e., the end of each epoch, we evaluate the performance of \( \theta \) in each language pair. For each language pair \( l \), if \( \theta \) outperforms \( \hat{\theta}_l \) in dev set \( D_l^{\text{dev}} \), \( \theta \) replaces \( \hat{\theta}_l \). To control whether to perform teaching in the current epoch, we define a set of distillation switches. In the validation stage, if the current model \( \theta \) outperforms the language-specific best checkpoint \( \hat{\theta}_l \), we turn off the distillation switch, not performing distillation in the next epoch. Conversely, the language-specific best checkpoint wins, we turn on the switch, performing distillation in the next epoch.

When the language-specific distillation switch is on, the parameters of the current model \( \theta \) is updated by optimizing both \( \mathcal{L}_{\text{NMT}} \) and \( \alpha \cdot \mathcal{L}_{\text{LSSD}} \). When the language-specific distillation switch is off, the model \( \theta \) is updated only by \( \mathcal{L}_{\text{NMT}} \), where \( \alpha = 0 \). The training loss is calculated as:

\[
\mathcal{L} = \mathcal{L}_{\text{NMT}} + \alpha \mathcal{L}_{\text{LSSD}}
\]

where the \( \alpha \) is the weight of distillation loss. And \( \mathcal{L}_{\text{LSSD}} \) is calculated as the cross-entropy between output distributions of \( \hat{\theta}_l \) and \( \theta \), which is formalized as:

\[
\mathcal{L}_{\text{LSSD}} = -\sum_{i \leq |y|} \sum_{w \in \mathcal{V}} P(w|x, y_{<i}; \hat{\theta}_l) \cdot \log P(w|x, y_{<i}; \theta)
\]

**3.3 Three Modes for LSSD**

To avoid the negative transferring (student better than teacher) in some translation instances, we devise three different instance-selection modes for LSSD: LSSD-WHOLE, LSSD-SELECTIVE and LSSD-ADAPTIVE. All of them could be generalized as multiplying the distillation loss with an instance-level weight:

\[
\mathcal{L} = \mathcal{L}_{\text{NMT}} + \alpha \mathcal{L}_{\text{LSSD}} \mathcal{G}
\]

where \( \mathcal{G} \) is the instance-level weight which is determined by the performance difference between teacher and student. And different modes correspond different implementation of \( \mathcal{G} \).
### Table 1: BLEU scores of baselines and our methods. The up section shows results reported in original papers. The bottom section shows results from the BASELINE implemented by us and three modes of LSSD. The highest BLEU value on each setting is in bold. “Related” and “Diverse” refer to the TED-8-Related dataset and the TED-8-Diverse dataset.

| Method                  | MANY-TO-ONE Related | Diverse | ONE-TO-MANY Related | Diverse |
|-------------------------|---------------------|---------|---------------------|---------|
| MultiDDS-S (Wang et al., 2020a) | 25.52               | 27.00   | 17.32               | 18.24   |
| MultiUAT (Wu et al., 2021)   | 26.39               | 27.83   | 18.64               | 19.76   |
| CCL-M (Zhang et al., 2021b)  | 26.73               | 28.34   | 18.89               | 19.53   |
| χ-IBR (Zhou et al., 2021)    | 28.71               | 29.74   | 22.21               | 23.44   |
| BASELINE               | 27.83               | 29.00   | 21.85               | 22.85   |
| LSSD-Whole             | 29.15               | 30.57   | 22.20               | 23.55   |
| LSSD-Selective         | 28.65               | 30.24   | 22.15               | 23.16   |
| LSSD-Adaptive          | **29.23**           | **30.59**| **22.09**           | **23.39**|

#### LSSD-WHOLE. In this mode, we statically perform distillation with the same instance-level weight for all data samples. In practice, it is equivalent to set $G = 1$.

#### LSSD-SELECTIVE. This mode is based on the idea that we only perform distillation in data samples where the teacher performs better than the student. And this is implemented as:

$$G(x, y, \theta, \hat{\theta}_l) = \begin{cases} 
0, & f(x, y, \hat{\theta}_l) < f(x, y, \theta) \\
1, & f(x, y, \hat{\theta}_l) \geq f(x, y, \theta)
\end{cases}$$  

(6)

where $f(x, y, \theta)$ is a score function to measure the performance of $\theta$ in sample $(x, y)$, which is adopted as the confidence (i.e., $P_\theta(y|x)$) in this work.

#### LSSD-ADAPTIVE. In this mode, we let the instance-level weight adaptively change with the performance difference, which is implemented as:

$$G(x, y, \theta, \hat{\theta}_l) = min\left(\frac{f(x, y, \hat{\theta}_l)}{f(x, y, \theta)}, \sigma\right)$$  

(7)

where $\sigma$ is used to truncate the $G$ when higher than $\sigma$, value of which is set to 2 simply.

### 4 Experiments

#### 4.1 Settings

**Datasets** Following previous works (Wang et al., 2020a; Zhou et al., 2021), we conduct experiments on two datasets:

**TED-8-Related**: a parallel corpus of 8 languages to (from) English. 4 low-resource languages (LRLs) (Azerbaijani: aze, Belarusian: bel, Glacial: glg, Slovak: slk) and a related high-resource language for each LRL (Turkish: tur, Russian: rus, Portuguese: por, Czech: ces).

**TED-8-Diverse**: a parallel corpus of another 8 languages to (from) English, which are picked without consideration for relatedness (Bosnian: bos, Marathi: mar, Hindi: hin, Macedonian: mkd, Greek: ell, Bulgarian: bul, French: fra, Korean: kor).

For each dataset, we experiment on two multilingual translation scenarios: 1) MANY-TO-ONE (M2O): translating 8 languages to English in this work; 2) ONE-TO-MANY (O2M): translating English to 8 different languages. Both of these two datasets have around 570k sentence pairs. The detailed data statistics are placed in Appendix A.

**Hyperparameters** We verify the effectiveness of LSSD in the Transformer (Vaswani et al., 2017) as implemented in fairseq (Ott et al., 2019) with 6 layers and 8 attention heads. And we use the same hyperparameters with the previous SOTA Zhou et al. (2021) to obtain a strong baseline. The only difference between with Zhou et al. (2021) is that we train all models for 300 epochs which is less than theirs\(^2\). We follow Wang et al. (2020a) to preprocess both datasets using sentencepiece (Kudo and Richardson, 2018) with a vocabulary size of $8K$ for

\(^2\)We also tried for more epochs, while no performance gains happened
Table 2: BLEU scores of the baseline and LSSD on the test set of TED-8-Related dataset. Bold values indicate the best performance for each language pair. The left four languages are low-resource, and the right four languages are high-resource.

| Method | aze | bel | glg | slk | tur | rus | por | ces | Avg. |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| **M2O** |     |     |     |     |     |     |     |     |     |
| BASELINE | 13.11 | 19.57 | 30.14 | 32.36 | 27.07 | 25.85 | 44.89 | 29.66 | 27.83 |
| LSSD | **14.49** | **21.5** | **31.66** | **33.57** | **28.37** | **26.95** | **46.21** | **31.07** | **29.23** |
| ∆ | +1.38 | +1.93 | +1.52 | +1.21 | +1.30 | +1.10 | +1.32 | +1.41 | +1.40 |
| **O2M** |     |     |     |     |     |     |     |     |     |
| BASELINE | 7.40 | 13.09 | 25.71 | 25.66 | 16.93 | 20.94 | 41.68 | 23.35 | 21.85 |
| LSSD | **7.72** | **13.63** | **25.91** | **26.34** | **17.25** | **21.44** | **41.80** | **23.50** | **22.20** |
| ∆ | +0.32 | +0.54 | +0.20 | +0.68 | +0.32 | +0.50 | +0.12 | +0.15 | +0.35 |

Table 3: BLEU scores of the baseline and LSSD on the test set of TED-8-Diverse dataset. Bold values indicate the best performance for each language pair. Languages are ordered increasingly by data size from left to right.

| Method | bos | mar | hin | mkd | ell | bul | fra | kor | Avg. |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| **M2O** |     |     |     |     |     |     |     |     |     |
| BASELINE | 25.78 | 11.25 | 24.30 | 33.38 | 38.40 | 39.34 | 40.43 | 19.15 | 29.00 |
| LSSD | **27.33** | **12.34** | **24.87** | **35.37** | **40.12** | **41.41** | **42.04** | **21.24** | **30.59** |
| ∆ | +1.55 | +1.09 | +0.57 | +1.99 | +1.72 | +2.07 | +1.61 | +2.09 | +1.59 |
| **O2M** |     |     |     |     |     |     |     |     |     |
| BASELINE | 17.04 | 4.96 | 15.99 | 25.34 | 33.27 | 36.31 | 40.43 | 9.08 | 22.85 |
| LSSD | **17.97** | **5.33** | **16.51** | **27.71** | **34.02** | **36.70** | **40.85** | **9.28** | **23.55** |
| ∆ | +0.93 | +0.37 | +0.52 | +2.37 | +0.75 | +0.39 | +0.04 | +0.20 | +0.70 |

Each language. The complete set of hyperparameters can be found in Appendix B. All models are trained on 8 V100. And the performance is evaluated with BLEU score using sacreBLEU (Papineni et al., 2002; Post, 2018). We set the distillation weight \( \alpha \) to 2.5 for M2O translation and 0.6 for O2M translation respectively.

**Baselines** We compare our method with the vanilla Transformer (i.e., BASELINE). The same model configuration and hyper-parameters are used for both BASELINE and our LSSD. We also list the results reported on several former works: 1) MultiDDS-S (Wang et al., 2020a) dynamically adjusts the sampling strategy over different languages according to gradients similarity. 2) MultiUAT (Wu et al., 2021) adjusts the sampling strategy with model’s uncertainty on different languages. 3) CCL-M Zhang et al. (2021b) incrementally adds new languages into the training set in a curriculum learning manner. 4) \( \chi \)-IBR Zhou et al. (2021) models the MNMT as a distributionally robust optimization problem and resolve it with iterated best response scheme. Note that all these methods are in parallel research lines with LSSD.

**4.2 Main Results**

We summarize main results into Table 1. As we can see, 1) comparing to the BASELINE, each mode of LSSD significantly outperforms the BASELINE in M2O and O2M, which verifies the effectiveness of our method; 2) in three modes of LSSD, LSSD-ADAPTIVE performs best in M2O and LSSD-WHOLE performs best in O2M; 3) comparing to previous works, LSSD achieves higher BLEU scores in most settings, which indicates the superiority of our method.

Next, we look closer at results on per-languages for the BASELINE and LSSD. We take LSSD-WHOLE and LSSD-ADAPTIVE as the representative of LSSD in M2O and O2M respectively as their superior. Results on the TED-8-Related and the TED-8-Diverse are shown in Table 2 and Table 3. As we can see, on both datasets and settings, LSSD consistently outperforms the BASELINE in all language pairs. The maximum improvement is up to 2.37 on a single language (eng \( \rightarrow \) mkd). Both low-resource and high-resource languages benefit from LSSD. It is observed that improvements in M2O (up to 1.59) are larger than which in O2M (up to 0.7). We will further discuss this in Section 5.2.

5 Analysis

5.1 Alleviating the Loss Gap

In this section, we analyze whether LSSD closes the loss gap between the overall best checkpoint and language-specific best checkpoints. To measure this gap, we calculate the total loss difference between them, which is referred to as “Distance to Upper-Bound (DUB)”. Formally, it is calculated

\[
\text{Distance to Upper-Bound (DUB)} = \sum_{i=1}^{n} (L_i - L_{\text{best}}(i))
\]

\( L_i \) is the loss of the overall best checkpoint, and \( L_{\text{best}}(i) \) is the loss of the language-specific best checkpoint.

In terms of average BLEU over both datasets and settings, LSSD-ADAPTIVE (26.33) > \( \chi \)-IBR (26.03).
Figure 3: Loss curves of the BASELINE and LSSD in M2O and O2M settings on TED-8-Diverse dataset. The left half part shows curves of M2O. The right half part shows curves of O2M. The first row shows low-resource language pairs. The second row shows high-resource language pairs. The third row shows average losses. The x-axis indicates training epochs. The y-axis indicates loss in dev sets. The solid lines are loss curves of our method. The dotted lines are loss curves of the baseline. Due to space limitation, we only display the loss curves for 4 language pairs (2 low-resource + 2 high-resource) in each setting.

Table 4: Distance to upper-bound in all settings for the BASELINE and LSSD.

| Method      | MANY-TO-ONE Related | Diverse | ONE-TO-MANY Related | Diverse |
|-------------|----------------------|---------|---------------------|---------|
| Baseline    | 0.57                 | 0.27    | 0.51                | 0.64    |
| LSSD        | 0.32                 | 0.21    | 0.30                | 0.15    |
| \(\Delta\)  | \(\downarrow 44\%\)  | \(\downarrow 22\%\) | \(\downarrow 41\%\) | \(\downarrow 77\%\) |

We first calculate the difference between losses of the overall best checkpoint \(\theta\) and language-specific best checkpoint \(\hat{\theta}_l\) in each language pair. And then accumulating these loss differences. We list the DUB for BASELINE and LSSD (LSSD-ADAPTIVE in M2O and LSSD-WHOLE in O2M) in Table 4. As observed, the loss gaps in our method are significantly lower than the baseline (decreased 46% on average).

5.2 Comparison of Loss Curves

In this section, we analyze different methods from the perspective of loss curves, as illustrated in Figure 3. First, we compare the loss curves of the BASELINE with LSSD (solid lines vs dotted lines). One important observation is that the BASELINE suffers from serious over-fitting in low-resources, which is caused by too long training time for low-resources. Due to knowledge distillation as regularization, LSSD effectively alleviates this problem by providing soft training targets (Hinton et al., 2015). Next, we compare the loss curves in M2O and O2M (left vs right half section), and we find that M2O suffers from more serious over-fitting than O2M. Particularly, the average loss of BASELINE in M2O begins increasing since about epoch 24, while the O2M begins increasing since about epoch 100. Therefore, M2O benefit more from LSSD than O2M. Last, we further make the comparisons among three variants of our proposed LSSD. In M2O, LSSD-ADAPTIVE performs better, while in O2M, LSSD-WHOLE performs better. Complete loss curves on both datasets and settings could be found in Appendix B.

5.3 Effect on Distillation Weight \(\alpha\)

As Equation 3 shows, we train MNMT models with NMT loss and the \(\alpha\)-weighted distillation loss jointly. And we demonstrate the effect on different \(\alpha\) for LSSD-Whole, which is shown in Figure 4. As we can see, \(\alpha = 2.0\) is best for M2O and \(\alpha = 0.6\) is best for O2M, which is significantly smaller than in M2O. To better understand this, we compare the loss curve in M2O with O2M (Figure 5 vs 6, Figure 7 vs 8) and find that O2M converges later than M2O (epoch 128 vs 66). It means that models in O2M might learn from immature teachers more times. Therefore, a smaller distillation strength is
more able to avoid this risk.

Figure 4: Effect of different distillation weights on LSSD. For simplicity, we take LSSD-WHOLE as the represent of LSSD.

6 Related Works

6.1 Advances in MNMT

Recently, MNMT mainly focus on: 1) designing effective parameters sharing strategy (Zhang et al., 2021a; Zhu et al., 2021; Xie et al., 2021; Lin et al., 2021); 2) obtaining language-agnostic representations (Zhu et al., 2020; Pan et al., 2021); 3) incorporating pre-training models (Siddhant et al., 2020; Wang et al., 2020b); 4) resolving the data imbalance among diverse languages (Wang et al., 2020a; Wu et al., 2021; Zhang et al., 2021b; Zhou et al., 2021). Different from them, LSSD is designed for bridging the gap between the overall best checkpoint and language-specific best checkpoints.

6.2 Knowledge Distillation in NMT

To the best of our knowledge, Kim and Rush (2016) first apply knowledge distillation in bilingual NMT and propose a sequence-level distillation. Wei et al. (2019) propose to avoid over-fitting by guiding the training process with best checkpoints. In MNMT, Tan et al. (2019) use knowledge distillation to close the gap between the MNMT model and bilingual models. Specifically, they train a bilingual model for each language and use these bilingual models as teachers to guide the MNMT model. However, their work is base on the hypothesis that there are enough training data for each language pair to train a promising bilingual teacher. In fact, most languages suffer from the resource-scarcity problem. In our work, LSSD’s teacher is the MNMT model, which has a better performance for low resource language translation via transfer learning. And we develop three different instance-selection modes for LSSD via weighing teacher and student performance. Those differences ensure that LSSD gains significant improvement in both low-resource and high-resource languages without extra monolingual/bilingual training data.

7 Conclusion

In this work, we propose a novel training strategy Language-Specific Self-Distillation (LSSD) to mitigate the gap between the overall best checkpoint and language-specific best checkpoints. Moreover, we devise three instance-selection modes for LSSD. Experimental results on two benchmarks demonstrate that LSSD achieves state-of-the-art performance. Through analysis experiments, we also find that: 1) LSSD significantly closes the performance gap (decreased 46% on average); 2) low-resource and high-resource languages equally benefit from LSSD.

References

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

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 Comput. Surv., 53(5).

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In ACL-IJCNLP.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. arXiv preprint.

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

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. TACL.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In EMNLP.

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.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. In ACL-IJCNLP.

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 NAACL.

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.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In ACL-IJCNLP.

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 on Association for Computational Linguistics, ACL ’02, page 311–318, 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.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In ACL.

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

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 ICLR.

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. 2020a. Balancing training for multilingual neural machine translation. In ACL.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020b. Multi-task learning for multilingual neural machine translation. In EMNLP.

Hao-Ran Wei, Shujian Huang, Ran Wang, Xin-yu Dai, and Jiajun Chen. 2019. Online distilling from checkpoints for neural machine translation. In NAACL.

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 EMNLP.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021. Importance-based neuron allocation for multilingual neural machine translation. In ACL-IJCNLP.

C. Yang, L. Xie, C. Su, and A. L. Yuille. 2019. Snapshot distillation: Teacher-student optimization in one generation. In CVPR.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021a. Share or not? learning to schedule language-specific capacity for multilingual translation. In ICLR.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In ICCV.

Mingliang Zhang, Fandong Meng, Yunhai Tong, and Jie Zhou. 2021b. Competence-based curriculum learning for multilingual machine translation. In Findings of EMNLP 2021.

Chunting Zhou, Daniel Levy, Xian Li, Marjan Ghazvininejad, and Graham Neubig. 2021. Distributionally robust multilingual machine translation. In EMNLP.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In ACL.

Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. 2021. Counter-interference adapter for multilingual machine translation. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2812–2823, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In NAACL.

A Data Statistics

Training set sizes of the TED-8-Diverse and TED-8-Related are listed in Tab. 5.
Figure 5: Loss curves in M2O translation on Diverse dataset for baseline and LSSD. The x-axis indicates training epochs. The y-axis indicates loss in dev sets. The 4 languages in the first row are low-resource languages. The 4 languages in the second row are high-resource languages. The average losses over all languages are drawn in the third row. The solid lines are loss curves of our method. The dotted lines are loss curves of the baseline. For clarity, we only show loss curves from epoch 10 to 100.

Figure 6: Loss curves in O2M translation on Diverse dataset for baseline and LSSD. The x-axis indicates training epochs. The y-axis indicates loss in dev sets. The 4 languages in the first row are low-resource languages. The 4 languages in the second row are high-resource languages. The average losses over all languages are drawn in the third row. The solid lines are loss curves of our method. The dotted lines are loss curves of the baseline. For clarity, we show the results from epoch 20 to 300.

B Hyperparameters

In this section, we list the details of hyperparameters we use for the experiments.

- We adopt the architecture with 6 layers and 8 attention heads.
- The embedding dimension is 512 and the FFN has a dimension of 2048.
- We 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).
- For regularization, we use the dropout as 0.3 (Srivastava et al., 2014) and the label smoothing as 0.1 (Szegedy et al., 2016).
- For sampling strategy, we use temperature-based sampling (Arivazhagan et al., 2019) and set $\tau = 1$.
- For inference, we use beam search with beam size 5.
| DIVERSE | RELATED |
|---------|---------|
| language | num | language | num |
| --- | --- | --- | --- |
| bos | 5,664 | bel | 4,509 |
| mar | 9,840 | aze | 5,946 |
| hin | 18,798 | glg | 10,017 |
| mkl | 25,335 | slk | 61,470 |
| ell | 134,327 | cse | 103,093 |
| fra | 192,304 | tur | 182,470 |
| bul | 174,444 | por | 184,755 |
| kor | 205,640 | rus | 208,458 |

Table 5: Data statistics for the TED-8-Diverse dataset and the TED-8-Related dataset. ‘num’ refers to the number of sentence pairs in the training set.

C Loss Curves

In this section, we show loss curves for the BASELINE and LSSD. The loss curve on TED-8-Diverse M2O translation is shown in Figure 5. The loss curve on TED-8-Diverse O2M translation is shown in Figure 6. The loss curve on TED-8-Related M2O translation is shown in Figure 7. The loss curve on TED-8-Related O2M translation is shown in Figure 8.

D Extra Training Overhead of LSSD

It is noteworthy that LSSD takes some extra training overhead. Taking the training in TED-8-Diverse M2O as an example, the BASELINE and LSSD-WHOLE spend 5.7 hours and 7.6 hours respectively. LSSD-SELECTIVE and LSSD ADAPTIVE spend 8 hours as taking more operations. In terms of GPU memory usage, LSSD takes up about extra 2.4GB of memory mainly for storing teacher models (each one takes about 300MB), which is affordable.
Figure 7: Loss curves in M2O translation on the Related dataset for baseline and LSSD. The x-axis indicates training epochs. The y-axis indicates loss in dev sets. The 4 languages in the first row are low-resource languages. The 4 languages in the second row are high-resource languages. The average losses over all languages are drawn in the third row. The solid lines are loss curves of our method. The dotted lines are loss curves of the baseline. For clarity, we show the results from epoch 10 to 100.

Figure 8: Loss curves in O2M translation on the Related dataset for baseline and LSSD. The x-axis indicates training epochs. The y-axis indicates loss in dev sets. The 4 languages in the first row are low-resource languages. The 4 languages in the second row are high-resource languages. The average losses over all languages are drawn in the third row. The solid lines are loss curves of our method. The dotted lines are loss curves of the baseline. For clarity, we show the results from epoch 30 to 300.