Adaptive Token-level Cross-lingual Feature Mixing for Multilingual Neural Machine Translation

Junpeng Liu[1] **Kaiyu Huang**[2] **Jiuyi Li**[1]
Huan Liu[1] **Jinsong Su**[3] **Degen Huang**[1]*
[1]Dalian University of Technology
[2]Institute for AI Industry Research, Tsinghua University [3]Xiamen University
{liujunpeng_nlp, lee.91, liuhuan4221}@mail.dlut.edu.cn
huangkaiyu@air.tsinghua.edu.cn
jssu@xmu.edu.cn huangdg@dlut.edu.cn

Abstract

Multilingual neural machine translation aims to translate multiple language pairs in a single model and has shown great success thanks to the knowledge transfer across languages with the shared parameters. Despite promising, this share-all paradigm suffers from insufficient ability to capture language-specific features. Currently, the common practice is to insert or search language-specific networks to balance the shared and specific features. However, those two types of features are not sufficient enough to model the complex commonality and divergence across languages, such as the locally shared features among similar languages, which leads to sub-optimal transfer, especially in massively multilingual translation. In this paper, we propose a novel token-level feature mixing method that enables the model to capture different features and dynamically determine the feature sharing across languages. Based on the observation that the tokens in the multilingual model are usually shared by different languages, we insert a feature mixing layer into each Transformer sublayer and model each token representation as a mix of different features, with a proportion indicating its feature preference. In this way, we can perform fine-grained feature sharing and achieve better multilingual transfer. Experimental results on multilingual datasets show that our method outperforms various strong baselines and can be extended to zero-shot translation. Further analyses reveal that our method can capture different linguistic features and bridge the representation gap across languages.[1]

1 Introduction

Multilingual neural machine translation (MNMT) (Ha et al., 2016; Johnson et al., 2017) handles several translation directions in a single model. These multilingual models have been shown to be capable of facilitating the knowledge transfer across different languages (Lakew et al., 2018; Tan et al., 2019; Zhang et al., 2020) and enabling translations between language pairs unseen in training (Johnson et al., 2017; Al-Shedivat and Parikh, 2019; Gu et al., 2019; Zhang et al., 2020). Due to the above advantages, MNMT is appealing and has drawn much attention in recent years.

The success of MNMT comes at the cost of insufficient ability to capture language-specific features (Zhang et al., 2021). Since the model parameters are shared across languages, the MNMT model tends to preserve the shared features but ignore the language-specific ones. Therefore, researchers resort to language-specific modeling to capture and balance those two types of features. Some works attempt to insert additional language-specific modules into the original MNMT model (Wang et al., 2019; Bapna and Firat, 2019; Zhang et al., 2020, 2021). However, those methods are sensitive to the structure and location of language-specific modules and require specialized manual design. To avoid this problem, other works turn to search language-specific networks in the MNMT model (Lin et al., 2021; Xie et al., 2021). Those methods generally adopt the multi-stage training strategy to find and fine-tune the language-specific parameters, which increases the training complexity, especially in massively multilingual translation settings.

Another pitfall of the above methods is that dividing the features into shared and language-specific ones may not be sufficient to model the complicated commonality and divergence across languages. Previous studies (Tan et al., 2019; Oncevay et al., 2020) have shown that similar languages generally share more commonality, and clustering them together can boost their translation performance. Moreover, Lin et al. (2021) also demonstrates that there are some overlaps between the language-specific networks of similar languages. These observations

*Corresponding Author
[1]Our code is available at https://github.com/raburabu91/HiTrans

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10097–10113
December 7-11, 2022

indicate that there are some locally shared features among similar languages which are important to the multilingual transfer. However, those features are not effectively used in the current language-specific models, which motivates us to model more fine-grained features of different languages to facilitate the multilingual transfer.

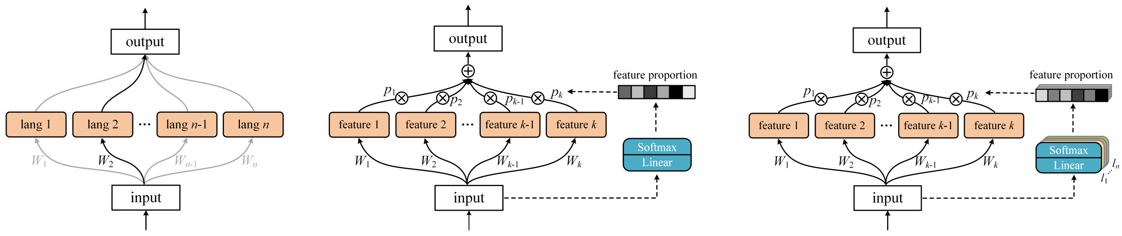
In this work, we propose a novel token-level cross-lingual feature mixing method that enables the model to adaptively determine the feature sharing during training. Based on the observation that the tokens in multilingual vocabulary are usually shared by different languages, we assume that each token representation contains a mix of lexical and linguistic features, with a feature proportion indicating its feature preference. Specifically, we employ a set of linear transformations to capture different features, on which we perform weighted feature aggregation with the specific feature proportion. By varying the feature proportions, we can retain the locally shared features and control the knowledge sharing across different languages. Our main contributions are summarized as follows:

- We propose a method that can perform fine-grained feature extraction and aggregation in the MNMT model without explicit shared and specific division, and can dynamically determine the feature sharing across languages with the adaptive feature proportions.
- We study the feature proportions and the representation space learned by our method, and find that our method can implicitly characterize a mix of linguistic features and narrow the representation gap across languages.
- We conduct extensive experiments on several multilingual datasets in different translation scenarios. Experimental results and in-depth analyses show that our method outperforms the language-specific models, especially in massively multilingual translation, and can be easily extended to boost zero-shot translation and alleviate the off-target issue.

2 Related Work

Our work closely relates to the language-specific modeling in MNMT. Early studies focus on increasing the shared parts of separate bilingual models for better knowledge transfer. These works include sharing encoders (Dong et al., 2015), sharing attention layers (Firat et al., 2016) and sharing decoders (Zoph and Knight, 2016). Later, Ha et al. (2016) and Johnson et al. (2017) develop a universal MNMT model with an artificial language token added to the source sentence to indicate the target language. While the share-all paradigm generally captures the commonality of languages but ignores the specific features of each language. To this end, researchers turn to language-specific modeling for better balance between sharing and specific, including redesigning parameter sharing strategies (Blackwood et al., 2018; Sachan and Neubig, 2018; Wang et al., 2019; Vázquez et al., 2019), training separate models for different language clusters (Tan et al., 2019), inserting lightweight adapters (Bapna and Firat, 2019), routing shared or language-specific path (Zhang et al., 2021), dividing general and specific networks or neurons (Lin et al., 2021; Xie et al., 2021) and parameter differentiation (Wang and Zhang, 2021). However, these methods do not make full use of the locally shared features across similar languages, leading to sub-optimal cross-lingual transfer, especially in massively multilingual translation. Instead, we propose a feature mixing method which is a variant of Mixture-of-Experts (MoE) models (Shazeer et al., 2017; Lepikhin et al., 2020). We discuss two gating mechanisms and analyze the impact of the location and sparsity of the MoE layer (CLM module) on multilingual translation performance.

Our work is also related to zero-shot translation. Some studies resort to forming language-agnostic representations. Arivazhagan et al. (2019a) and Pham et al. (2019) introduce auxiliary training objectives to align the representations of different languages. Pan et al. (2021) bridges the cross-lingual representations with additional dictionary and contrastive learning. Liu et al. (2021) disentangles the positional information by relaxing the structural constraint. Other studies explore to enhance the language-specific features in translation. Wang et al. (2019) and Yang et al. (2021) employ an additional target language prediction task to train the model to distinguish different languages. Philip et al. (2020) adopt monolingual adapter to model the language-specific features. Our work continues in these directions, but with a special focus on combining different feature mixing models in the encoder and decoder to build a language-agnostic encoder and language-aware decoder.



(a) Language-specific model (b) *sCLM* model (c) *mCLM* model

Figure 1: Comparison of language-specific, *sCLM* and *mCLM* model. The residual connection and layer normalization are not visualized here for brevity.

3 Method

Our main idea is to model the commonality and divergence of different languages in a fine-grained way to retain more shared features, especially those locally shared by similar languages to facilitate the multilingual transfer. To achieve this, each language is considered to contain a mix of different features rather than solely the shared and specific ones, as shown in Figure 1. Specifically, we first project each token representation into different subspaces with a set of linear transformations to capture different features and calculate the corresponding feature proportion based on the token representation itself. Then we take the weighted averaging of different linear transformations as the feature-mixed representation. The proportion indicates the importance of each feature and determines the knowledge sharing across different languages.

3.1 Feature Proportion

Our proposed method is motivated by the observation that the token (e.g. word or subword) in the multilingual vocabulary usually contains several different lexical and linguistic features. On the one hand, a token shared by different languages naturally embodies different lexical and semantic meanings. On the other hand, a token also contains various contextual and structural information because its representation is essentially learnt from all the tokens in the sentence. Inspired by Jiang et al. (2020), we assume that each token holds a mix of those lexical and linguistic features with a certain proportion indicating its feature preference in different languages. Specifically, given a token representation $x \in \mathbb{R}^d$ and k features, we parameterize the feature proportion $\mathcal{P}(x)$ with a linear transformation followed by a softmax function. We also add a smoothing parameter α to prevent the output $\mathcal{P}(x)$ from collapsing towards 0 or 1:

$$\mathcal{P}(x) = (1-\alpha) \cdot \mathrm{softmax}(xP) + \alpha/k \qquad (1)$$

where $P \in \mathbb{R}^{d \times k}$ is the feature projection weight, $\alpha \in (0, 1)$ smooths the probability so as to activate all the features.

3.2 Adaptive Token-level Feature Mixing

Previous studies (Bapna and Firat, 2019; Zhang et al., 2020, 2021) employ individual parameters for each language pair to capture the language-specific features. However, those methods are weak in their ability to capture the locally shared features among similar languages. To solve this problem, we take the weighted aggregation of different features based on a specific proportion $\mathcal{P}(x)$ as the language-specific representations. In this way, the feature sharing across different languages can be controlled by varying their feature proportions. Specifically, we consider linear transformations $\{W_j\}_{j=1}^{k}$ for k features on the i-th input token representation h_i, the weighted aggregation of linear transformations can be written as follows:

$$\tilde{h}_i = \sum_{j=1}^{k} h_i W_j \cdot \mathcal{P}_j(h_i) \qquad (2)$$

where W_j is the linear transformation used to model the j-th feature and $\mathcal{P}_j(h_i)$ denotes the proportion on the j-th feature for representation h_i.[2] In multilingual translation, the token representations in each source input naturally contain the target language information since a target language token is added to the source sentence. This indicates the feature proportions of the same token can also be different when translated into different languages.

[2] To make the number of parameters manageable, we separately maintain a set of linear transformations in the encoder and the decoder, and share them across all the encoder or decoder sublayers.

This property makes our method more flexible to capture the specific features in different conditions.

Our feature mixing method can be seen as a heuristic variation of Mixture-of-Experts (MoE) models (Shazeer et al., 2017; Lepikhin et al., 2020). However, compared to previous MoE models which are the sparse combination of the gating mechanism, we adopt a soft and smoothed gating network to retain all the potential shared features and replace the non-linear experts with linear ones for lower memory cost and fast training speed.

3.3 Cross-lingual Mixing Model

Based on the token-level feature mixing strategy, we introduce our cross-lingual mixing (CLM) module and its implementation in Transformer. Given the input representation h, CLM calculates the feature proportion $\mathcal{P}(h)$ and the weighted averaging representation $\tilde{h}$ as Equations 1 and 2. To make our CLM module optional and plug-able into any part of the Transformer network, we apply a residual connection followed by layer normalization (LN). The CLM module is finally formulated as follows:

$$z = \mathrm{LN}(h + \tilde{h}) \tag{3}$$

Since the tokens have different representations at each Transformer sublayer, their corresponding feature proportions are also different. To this end, we inject CLM modules into each sublayer and distinguish the feature projection weight P across different Transformer layers. Considering that the token may have various feature proportions in different languages, we propose two variants of CLM model according to the feature projection weight settings:

sCLM shares a single feature projection weight $P_s \in \mathbb{R}^{d \times k}$ across all the language pairs. This strategy may ease the proportion allocation in our method as it is highly input dependent.

mCLM employs a set of language-specific feature projection weights $\{P_m \in \mathbb{R}^{d \times k}\}_{m=1}^{N}$ for different language pairs. Although this strategy involves more parameters than *sCLM*, we hope that different proportion weights will make it more flexible in proportion allocation.

4 Experiments

4.1 Datasets

We evaluate our method in English-to-many and many-to-English translation scenarios. We also extend our method to zero-shot translation based on the observations in English-centric translation. For en-xx and xx-en translation, we test our method on the OPUS-100 and WMT benchmarks. For zero-shot translation, we evaluate our method on three datasets: IWSLT-17, Europarl and WMT-5. The detailed data descriptions are listed in Appendix A.1. We apply byte pair encoding (BPE) algorithm (Sennrich et al., 2016) using SentencePiece (Kudo and Richardson, 2018)[3] to preproess multlingual sentences with a joint vocabulary of 64K for OPUS-100/WMT-14 and 32K for IWSLT-17/Europarl/WMT-5.

4.2 Baselines

To make our evaluation convincing, we re-implement the original MNMT model and several previous works for comparison.

Multilingual (Johnson et al., 2017) The unified model which handles multiple languages in a single encoder-decoder model by adding a special language token to the source sentence.

+Adapter (Bapna and Firat, 2019) A set of light-weight adapters are injected into the vanilla MNMT model. The dimension of the projection layer is set to 128 and we train the model from scratch.

+CLSR (Zhang et al., 2021) This method employs a series of hard binary gates conditioned on token representations to dynamically choose the shared and language-specific paths.

Deep Transformer (Zhang et al., 2020) This method improves the model capacity by increasing the model depth to build a strong baseline. For fair comparisons, the model depth (for both encoder and decoder) are set to 26 and 8 for OPUS-100 and WMT-14, respectively.

4.3 Training and Evaluation

We employ Transformer-Base setting (Vaswani et al., 2017) in all our experiments on the open-source Fairseq implementation (Ott et al., 2019)[4]. The detailed model settings are in Appendix B. We insert the CLM modules into both encoder and decoder for en-xx translation but decoder only for xx-en translation based on the ablation study in Section 4.4.

We report the detokenized case-sensitive BLEU offered by SacreBLEU (Post, 2018)[5]. Following Zhang et al. (2021), we split the language

[3] `https://github.com/google/sentencepiece`

[4] `https://github.com/pytorch/fairseq`

[5] Signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.1.

Model	Model Size	en-xx					xx-en				
		Low	Med	High	All	WR	Low	Med	High	All	WR
Multilingual	76.96M	26.54	25.72	20.89	24.38	–	33.42	32.87	28.91	31.73	–
+Adapter	224.81M	+2.64	+3.25	+2.67	+2.85	93.62	+1.37	+2.36	+1.60	+1.78	88.30
+CLSR	136.08M	+2.35	+2.29	+1.73	+2.12	94.68	+1.64	+1.14	+0.98	+1.25	88.30
Deep Transformer	224.09M	+3.50	**+4.58**	**+3.49**	**+3.86**	**96.81**	+1.51	+1.77	**+3.66**	+2.31	86.17
sCLM$^\diamondsuit$	225.49M	**+3.56**	+4.33	+3.13	+3.67	**96.81**	+2.34	**+2.56**	+2.56	**+2.49**	**97.87**
mCLM$^\diamondsuit$	224.63M	+2.43	+3.79	+2.82	+3.01	94.68	**+2.61**	+2.14	+1.68	+2.14	92.55

Table 1: Translation quality for en-xx and xx-en on the OPUS-100 dataset. *sCLM*$^\diamondsuit$ and *mCLM*$^\diamondsuit$ represent the best *sCLM* and *sCLM* model, respectively. To match Adapter in parameters, the feature number k in *sCLM*$^\diamondsuit$ is 280/560 for en-xx/xx-en translation, while 194/388 in *mCLM*$^\diamondsuit$. Best results are highlighted in **bold**.

Model	Model Size	en-xx					xx-en				
		Low	Med	High	All	WR	Low	Med	High	All	WR
Multilingual	76.91M	16.35	19.05	25.07	20.32	–	23.08	25.67	27.70	25.46	–
+Adapter	97.37M	**+0.89**	**+1.06**	+1.12	**+1.03**	**100.0**	+0.23	+0.66	+0.39	+0.39	76.92
+CLSR	93.75M	+0.44	+0.52	+0.64	+0.54	**100.0**	+0.17	+0.56	+0.33	+0.32	92.31
Deep Transformer	91.63M	+0.63	+1.06	+1.17	+0.79	**100.0**	+0.68	+0.80	+0.17	+0.54	76.92
sCLM$^\diamondsuit$	95.49M	+0.85	+0.86	+1.01	+0.92	**100.0**	**+1.00**	**+1.11**	**+0.84**	**+0.96**	**100.0**
mCLM$^\diamondsuit$	98.09M	+0.76	+1.00	**+1.22**	+1.00	**100.0**	+0.58	+0.99	+0.68	+0.71	**100.0**

Table 2: Translation quality for en-xx and xx-en on the WMT-14 dataset. The feature number k in the two CLM models are 35/70 for en-xx/xx-en translation. Best results are highlighted in **bold**.

pairs in OPUS-100 and WMT-14 into three groups (Low/Med/High) according to their data size. We report the average BLEU for each group and Win Ratio (WR) indicating the proportion of language pairs on which our method beats the original MNMT model. In zero-shot translation, we also report the off-target rate to measure the accuracy of translating into the right target language.

4.4 Results

Results on OPUS-100. The results are summarized in Table 1. The comparisons between the multilingual baseline and our method suggest that the two variants of the CLM model can improve translation performance for both en-xx and xx-en directions in most language pairs (up to +3.67 BLEU & 96.81 WR on en-xx and +2.49 BLEU & 97.87 WR on xx-en). Moreover, our *sCLM*$^\diamondsuit$ also yields competitive results to the strong baseline with deeper architecture. Compared to +Adapter, our *sCLM*$^\diamondsuit$ and *mCLM*$^\diamondsuit$ achieve better translation performances and WR scores with similar parameters. The results show that adding an adapter module to capture language-specific features may not be sufficient in massively multilingual settings. Compared with +CLSR, our method also performs better, showing that the feature mixing strategy is more efficient than directly modeling and balancing the shared and language-specific features of different language pairs.

Results on WMT-14. The results are summarized in Table 2. Similar to Table 1, our method exceeds the multilingual baseline in all language pairs and beats the Deep Transformer model, confirming the effectiveness of our method. One noticeable difference is that the improvements on xx-en translation brought by +Adapter and +CLSR are not large. By contrast, our method achieves more remarkable BLEU gains and 100% WR scores. Another difference is that our method does not surpass +Adapter on en-xx directions. We ascribe this to the smaller number of similar language pairs in WMT-14, where the feature mixing may cause interference across languages, leading to performance degradation in some language pairs.

Ablation Study. To study the efficacy of each component in the CLM module, we evaluate models of different settings on the OPUS-100 dataset. The results are summarized in Table 3 and we make the following observations:

- When removing the gating mechanism from CLM modules, the language-specific model *LS* fails to surpass the multilingual baseline in

Model	Enc	Dec	Model Size	en-xx					xx-en				
				Low	Med	High	All	WR	Low	Med	High	All	WR
Multilingual			76.96M	26.54	25.72	20.89	24.38	–	33.42	32.87	28.91	31.73	–
LS	✓	✓	126.25M	-1.91	+0.02	-0.19	-0.69	37.23	-1.46	-1.38	-0.93	-0.92	23.70
sCLM	✓	✓	126.85M	**+2.59**	+2.44	+1.92	+2.32	**96.81**	+0.17	+0.27	+1.66	+0.70	75.75
sCLM-E	✓		101.90M	+0.48	+0.99	+1.02	+0.83	84.04	+1.79	+1.16	+1.10	+1.35	**96.81**
sCLM-D		✓	101.90M	+0.63	+1.10	+1.05	+0.92	86.17	-1.15	-0.79	+0.71	-0.41	59.57
mCLM	✓	✓	180.56M	+2.02	**+3.22**	**+2.49**	**+2.58**	94.68	+1.53	**+1.80**	**+1.97**	**+1.77**	88.30
mCLM-E	✓		128.75M	+1.33	+1.87	+1.65	+1.62	90.43	**+1.83**	+1.65	+1.28	+1.59	91.49
mCLM-D		✓	128.75M	+1.45	+1.96	+1.63	+1.68	88.30	+0.26	+0.68	+0.87	+0.60	78.72
Dedicated	✓	✓	153.70M	+1.93	+2.81	+2.17	+2.30	90.43	+1.01	+1.45	+1.80	+1.42	85.11

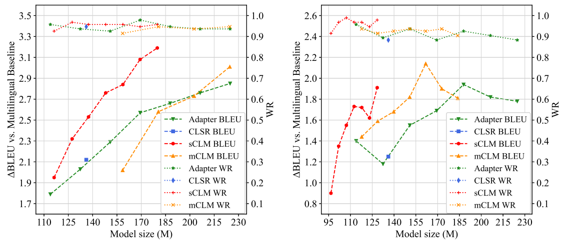
Table 3: Ablation study on OPUS-100 dataset. "✓" denotes the corresponding CLM modules are inserted in the encoder or the decoder. "*LS*": a language-specific model which removes the gating mechanism from CLM modules and makes the linear transformations $\{W_j\}_{j=1}^{k}$ language-specific. Specially, we keep the number of features and languages the same. "*Dedicated*": the combination of *sCLM*-E and *mCLM*-D. Best results are highlighted in **bold**.

most language pairs. The performance difference between *LS* and +Adapter shows that the structure and location of the language-specific modules have a large impact on the translation performance and the gating mechanism is important to mitigate the performance decline.

- For en-xx translation, the CLM modules are important to both the encoder and the decoder, while for xx-en translation, it tends to bring better performances when the CLM modules are only inserted into the encoder.

- Replacing the shared feature projection weight P_s with language-specific ones P_m (*sCLM* vs. *mCLM*) can further enhance the translation quality, especially on xx-en translation. We conjecture that the xx-en translation shares the same target language (English), so it is hard for *sCLM* to capture the specific characteristics of each language pair with the shared proportion weight, as the feature proportions are similar to each other. By contrast, *mCLM* employs different projection weights for each language pair, making it more flexible to model the differences across language pairs. The performance of the *Dedicated* model to some extent proves our conjecture.

More Comparisons. To further illustrate the superiority of our method, we quantify the trade-off between adapter/CLM capacity and performance gains on the OPUS-100 dataset.[6] The results are depicted in Figure 2. We also plot CLSR in the figure for a comprehensive comparison. *sCLM* consistently outperforms Adapter and CLSR on both en-xx and xx-en translations under the similar number of parameters. Moreover, *sCLM* achieves the best results with 20%-30% parameter reduction compared with Adapter. While *mCLM* only shows its superiority on xx-en translation due to the increased parameters. We also compare the decoding speed of each method in Appendix C.1.



(a) en-xx translation (b) xx-en translation

Figure 2: Comparisons of Adapter, CLSR, *sCLM* and *mCLM* under different model sizes.

5 Analysis

5.1 Feature Proportion Similarity

In our method, each token representation is encoded by aggregating all the features with a specific proportion. We explore whether CLM learns to allocate those feature proportions according to linguistic characteristics or not. We study the proportion allocation of *sCLM* for en-xx translation on the OPUS-100 testset. Specifically, we calculate the cosine similarity of different language pairs with their average token-level feature proportions (ATP) in both the encoder and the decoder. For

[6]The adapter capacity is changed by varying the bottleneck dimensions in the range of $D_A = \{32, 48, 64, 80, 96, 112, 128\}$, while the CLM capacity is changed by varying the number of features k in CLM modules in the range of $N_F = \{74, 94, 114, 134, 154, 174, 194\}$.

lang	it		ru		hi		tr	
	enc	dec	enc	dec	enc	dec	enc	dec
1	es	pt	uk	uk	ur	ne	ko	ja
2	pt	ca	mk	bg	ta	mr	ja	tk
3	fr	es	sk	be	ug	gu	ml	eo
4	gl	gl	de	mk	tg	cs	bs	et
5	ca	fr	bg	ky	bn	si	pl	uz

Table 4: Languages with top-5 similar ATP vector.

instance, given the testset of language pair l, $\mathcal{D}_l$, the ATP in the encoder is formulated as follows:

$$\text{ATP}_e^l = \frac{\sum_{X \in \mathcal{D}_l} \mathcal{P}_e}{\sum_{X \in \mathcal{D}_l} |X||\mathcal{N}_{enc}|} \tag{4}$$

where $|X|$ is the length of the input sentence X, $\mathcal{N}_{enc}$ represents the set of all the CLM modules in the encoder, and $\mathcal{P}_e$ denotes the total feature proportion of all the tokens in sentence X, which is given by $\mathcal{P}_e = \sum_{x \in X} \sum_{m \in \mathcal{N}_{enc}} \mathcal{P}_m(x)$. For each language pair l, we select the languages with the top-5 cosine similarity. Results for several languages are presented in Table 4 (see Appendix C.2 for full results) and we have two major findings:

- ***sCLM* captures the relationship in the language branch well.** As shown in Table 4, for languages from branches such as Romance (It) and Slavic (Ru), their most similar languages generally come from the same language branch. These results show that *sCLM* can implicitly capture not only the similarities between languages but also the differences among language branches despite they all belong to the Indo-European family. Moreover, languages from the same branch differ in their similar languages, suggesting that *sCLM* can characterize the specific features of languages by varying their feature proportions.

- ***sCLM* can also capture the word order divergence.** The dominant word order for most languages in our experiments is SVO, while for languages such as from Indic (Hi) or Turkic (Tr) branch, SOV is usually the dominant type. As shown in Table 4, *sCLM* selects those of the same word order (SOV) as their most similar languages despite they belong to different language families or even do not share the same scripts. For example, the most similar language for Tr (Turkish) in the encoder and decoder are Ko (Korean) and Ja (Japanese), respectively. Another explanation for this result is that those three languages are all exclusively concatenative languages.

In addition to the above findings, we also observe that *sCLM* can capture regional and cultural influences. For example, Ms (Malay), Id (Indonesian) and Vi (Vietnamese) share more similarities because they are close to each other in geographical location. Zh (Chinese) and Ja (Japanese) are more similar in the decoder due to cultural influences. These observations show that *sCLM* can characterize complex relationships across languages and fuse those information together well.

5.2 Representation Analyses

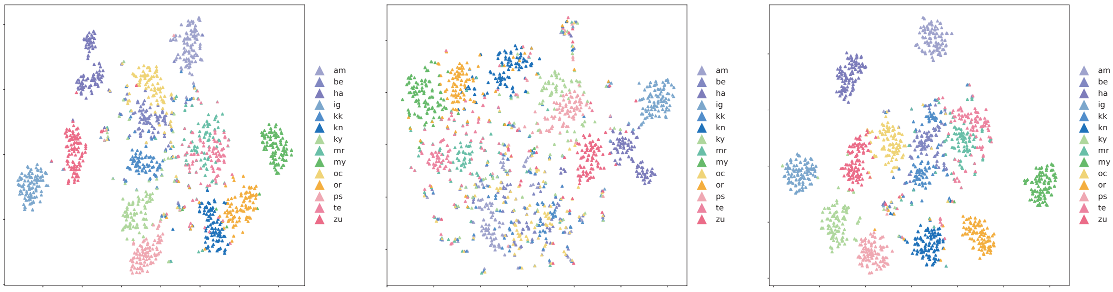
To interpret the superiority of our method over baselines, we delve into the encoder representations incurred by models on xx-en translations. We first employ the accuracy of similarity search tasks as a quantitative indicator of cross-lingual representation alignment following Pan et al. (2021), and then we visualize some sentence representations for further study and comparison.

5.2.1 Similarity Search

The data computing representations come from TED (Qi et al., 2018) and Flores (Goyal et al., 2021) as they provide multi-way translations in which sentences from each language are semantically equivalent to each other. For TED, we construct a multi-way parallel testset of 2296 samples covering 15 languages. For Flores, we select the first 100 sentences from each language resulting in a multi-way testset of 75 languages. The detailed descriptions of the two testsets are presented in Appendix A.2.

We conduct experiments in both English-Centric and Zero-Shot scenarios, and report the average top-1 accuracy of sentence similarity research on each dataset. The sentence representations are calculated by averaging the encoder outputs. The results are listed in Table 5.

English-Centric: Since English has never been seen by the encoder for xx-en translation, there is no available projection weight for *mCLM* to encode English sentences. Therefore, we only show the results of *sCLM* in this scenario. Our *sCLM* achieves notable accuracy improvements on both TED and Flores testset, suggesting that *sCLM* generalizes well to English with the shared projection weight and narrows the representation gap between



(a) Multilingual baseline (b) *sCLM* (c) *mCLM*

Figure 3: t-SNE visualizations of the encoder representations of 14 low-resource languages on xx-en translation encoded by Multilingual baseline, *sCLM* and *mCLM*.

Model	English-Centric		Zero-Shot	
	TED	Flores	TED	Flores
Multilingual	20.5%	39.1%	80.5%	74.8%
sCLM	**36.4%**	**58.3%**	**84.8%**	**80.0%**
mCLM	–	–	84.2%	75.1%

Table 5: The averaged sentence similarity search top-1 accuracy on TED and Flores testsets in **English-Centric** and **Zero-Shot** scenarios.

English sentences and their semantic equivalents in other languages.

Zero-Shot: The overall accuracy follows the rule that Multilingual < *mCLM* < *sCLM*, showing that the two proposed models can boost the cross-lingual representation alignment. One noticeable observation is that the improvements of *mCLM* on Flores are not as large as those on TED. We further visualize the sentence representations to explain this point and study the differences between the two proposed models.

5.2.2 Visualization and Comparison

To further study representation space learned by our *sCLM* and *mCLM*, we visualize the encoder representations on xx-en translation by reducing the 512-dim representations to 2-dim with t-SNE (Van der Maaten and Hinton, 2008). We use Flores devtest dataset for visualization as it covers languages of different data sizes. For clarity, we split the 74 non-English languages into three groups (Low/Med/High). We also visualize the representations of the multilingual baseline for comparison. The visualizations on low-resource languages are depicted in Figure 3 and the results on med- and high-resource languages are presented in Appendix C.3. We make the following observations:

- For the baseline model, most sentences from high-resource languages are clustered to their semantic equivalents in other languages while med-resource especially low-resource languages possess their own distinct clusters.
- For *sCLM*, sentences from low- and med-resource languages start to be assigned to their semantic clusters and the clustering results on high-resource languages are better than the multilingual baseline.
- For *mCLM*, it strengthens the trend that sentences from low-resource languages incline to form their individual clusters, despite the better clustering results in high-resource languages. These observations may explain the improvement gaps between TED and Flores (3.7% vs. 0.3%) in Zero-Shot scenario in Table 5 since all the languages in TED are high-resource.

These observations show the differences between our *sCLM* and *mCLM* models. *sCLM* improves the translations in the sense that it bridges the representation gap across languages while *mCLM* maps the representations of different languages into distinct subspaces, especially for low-resource languages. We argue that the representations learned by *sCLM* are more appealing as it clusters sentences based on their semantic similarities. Compared to high-resource languages, the representations in low- and med-resource languages are still not clustered well which need further research.

5.3 Extension to Zero-shot Translation

Recent studies (Arivazhagan et al., 2019a; Liu et al., 2021) show that zero-shot translation can

Dataset	Pivot	Multilingual	+*sCLM*-E	+*mCLM*-D
IWSLT-17	19.80	15.28 (7.23)	17.68 (5.48)	18.77 (2.46)
Europarl multiway	24.01	20.76 (0.78)	22.79 (0.51)	22.94 (0.50)
Europral w/o overlap	26.84	23.51 (0.67)	25.64 (0.52)	25.68 (0.46)
Europarl full	28.76	27.32 (0.51)	28.17 (0.49)	28.10 (0.49)
WMT-5	14.70	5.41 (51.0)	6.12 (48.4)	9.17 (25.0)

Table 6: Translation results on zero-shot directions. The average off-target rates (%) calculated by off-the-shelf LangID model from FastText (Joulin et al., 2016) are reported in brackets.

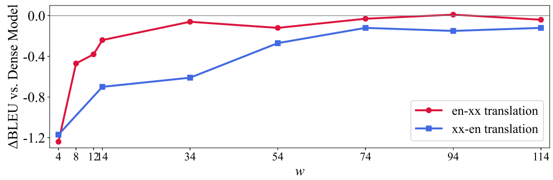


Figure 4: ΔBLEU score along with the increase of w in en-xx and xx-en translation on OPUS-100 dataset.

be boosted by facilitating the encoder to learn language-agnostic representations. Based on the observations in Section 5.2, we apply the CLM models to zero-shot translation. Specifically, we insert *sCLM* into the encoder to encourage the language-independent representations. Moreover, we also use *mCLM* to enhance the ability to distinguish different target languages in the decoder. In Table 6, our method substantially improves zero-shot translation quality and reduces the off-target translations even in the very challenging case of WMT-5, where languages are from different language branches and do not share scripts. In addition, our method also shows competitive results to the pivot models via English. These results demonstrate the strong transfer ability of our method.

5.4 About Sparsity

To verify whether all the features are essential to the representations, we study the sparsity by selecting the top-w important features for each token representation and pruning others. The performance of *sCLM* with different w are plotted in Figure 4. The performance on en-xx translation remarkably degrades only when $w < 14$, suggesting that some features are not important to the translation quality and can be pruned. Similar results can also be observed on xx-en translation. However, the degradation comes earlier ($w < 54$) than en-xx translation, showing that *sCLM* is more sensitive to the sparsity on xx-en translation.

6 Conclusion

In this paper, we propose a token-level cross-lingual feature mixing method that can capture different features and dynamically determine the feature sharing across languages. We employ a set of linear transformations to capture different features and aggregate them with specific proportions for each token representation. In this way, we can perform fine-grained feature sharing and achieve better multilingual transfer. Experimental results on multilingual datasets show that our method outperforms various strong baselines and can be extended to zero-shot translation. Further analyses reveal that our method can capture several different linguistic features and bridge the representation gap across languages. In future work, we plan to further study how to narrow the representation gap across low-resource languages for better translation performance and knowledge transfer.

Limitations

Despite effective, our method has the following limitations. An obvious limitation is that we employ additional parameters to model different features to ease the implementation of our method in massively multilingual translation. However, it increases the training cost and slows down the decoding speed. Another limitation is that although our method can bridge the representation gap across languages, the sentence representations in low-resource language still incline to possess their distinct clusters. In the future, we plan to improve the representation space of low-resource languages in the multilingual translation.

Acknowledgements

We sincerely thank all the anonymous reviewers for their insightful comments and suggestions to improve the paper. This work was supported by the National Key Research and Development Program of China (2020AAA0108004) and the National Natural Science Foundation of China (No.U1936109).

References

Maruan Al-Shedivat and Ankur Parikh. 2019. Consistency by agreement in zero-shot 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

1184–1197, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019a. The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019b. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Graeme Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3112–3122, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

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.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *CoRR*, abs/2106.03193.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of IWSLT 2016*.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1834, Online. 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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 641–652, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 293–305, Online. Association for Computational Linguistics.

Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2021. Improving zero-shot translation by disentangling positional information.

In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1259–1273, Online. Association for Computational Linguistics.

Arturo Oncevay, Barry Haddow, and Alexandra Birch. 2020. Bridging linguistic typology and multilingual machine translation with multi-view language representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2391–2406, Online. Association for Computational Linguistics.

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.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alexander Waibel. 2019. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 13–23, Florence, Italy. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. 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.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. Association for Computational Linguistics.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Raúl Vázquez, Alessandro Raganato, Jörg Tiedemann, and Mathias Creutz. 2019. Multilingual NMT with a language-independent attention bridge. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 33–39, Florence, Italy. Association for Computational Linguistics.

Qian Wang and Jiajun Zhang. 2021. Parameter differentiation based multilingual neural machine translation. *arXiv preprint arXiv:2112.13619*.

Yining Wang, Long Zhou, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2019. A compact and language-sensitive multilingual translation method. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1213–1223, Florence, Italy. Association for Computational Linguistics.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021. Importance-based neuron allocation for multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for*

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 5725–5737, Online. Association for Computational Linguistics.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021. Improving multilingual translation by representation and gradient regularization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7266–7279, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations 2021*.

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

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

A Dateset Details

A.1 Training Data

We perform en-xx and xx-en translations on the OPUS-100 and WMT-14 benchmarks, and zero-shot translations are evaluated on IWSLT-17, Europarl and WMT-5 datasets. We give detailed descriptions of these dataset used in this work.

OPUS-100. We collect 94 language pairs from (Zhang et al., 2020)'s release[7] by discarding those without valid/test sets. We use the official valid/test sets for evaluation.

WMT-14. We use the same training/valid/test sets as Zhang et al. (2021) except that we limit the training sentence pairs in each direction to 10M by random sampling.

IWSLT-17. We select 3 language pairs (En $\leftrightarrow$ $\{$It, Nl, Ro$\}$) from the official dataset[8], and perform 6 zero-shot translations between the 3 non-English langauges. The datasets are described in Table 7.

Europarl. We use the training/valid/test datasets released by (Liu et al., 2021) and conduct experiments under three conditions following Liu et al. (2021).

WMT-5. We collect 4 language pairs from WMT-14: En-De (4.5M), En-Hi (0.3M), En-Ru (10M) and En-Zh (10M). We study this challenging case where the training data is imbalanced and the languages involved in zero-shot directions are different in scripts. We evaluate the zero-shot performance on the Flores devtest which contains 1012 sentences in each direction.

A.2 Evaluation Data

We employ the TED testset and Flores devtest for representation analysis in Section 5.2, and we give more detailed descriptions.

TED. We construct a multi-way parallel testset of 2296 samples covering 15 languages including Arabic, Czech, German, English, Spanish, French, Italian, Japanese, Korean, Dutch, Romanian, Russian, Turkish, Vietnamese and Chinese. Note that the languages in TED are all high-resource in the OPUS-100 dataset.

[7] `https://object.pouta.csc.fi/OPUS-100/v1.0/opus-100-corpus-v1.0.tar.gz`

[8] `https://sites.google.com/site/iwsltevaluation2017`

Language Pair	train	valid	test
En-It	231.6K	929	1566
En-Nl	237.2K	1003	1777
En-Ro	220.5K	914	1678
It-Ro	217.5K	914	1643
Nl-Ro	206.9K	913	1680
It-Nl	233.4K	1001	1669

Table 7: Statistics of IWSLT-17 dataset.

	Language
Low	am, be, ha, ig, kk, kn, ky, mr, my, oc, or, ps, te, zu
Med	af, as, az, cy, ga, gl, gu, hi, ka, km, ku, ml, ne, pa, ta, tg, ur, uz, xh
High	ar, bg, bn, bs, ca, cs, da, de, el, es, et, fa, fi, fr, he, hr, hu, id, is, it, ja, ko, lt, mk, ms, mt, nl, no, pl, pt, ro, ru, sk, sl, sr, sv, th, tr, uk, vi, zh

Table 8: Languages in Flores devtest set used for similarity search.

Flores. For Flores, we select the first 100 sentences from the devtest for each language resulting in a multi-way testset of 75 languages. We split the languages into three groups (Low/Med/High) according to their data size in the OPUS-100 dataset. The detailed statistics are listed in Table 8.

B Implementation Details

For fair comparison, we employ Transformer base in all our experiments, which consists 6 stacked encoder/decoder layers and 8 attention heads, with the model size d_{model} of 512 and feed-forward dimension d_{ffn} of 2048.

For model training, we use the temperature-based sampling strategy to balance the training data distribution with a temperature of $T = 5$ (Arivazhagan et al., 2019b), and set *share-all-embeddings* in Fairseq to save parameters. All the model parameters are optimized using Adam optimizer (Kingma and Ba, 2014) ($\beta_1 = 0.9, \beta_2 = 0.98$) with label smoothing of 0.1. The learning rate is scheduled as Vaswani et al. (2017) with a warm-up step of 4000 and a peak learning rate of 0.0005. The dropout rate is set to 0.1 and the smoothing parameter α in Equation 1 is set to 0.05. We train all models with a batch of 4096 and set *update_freq* in Fairseq to 4. The training sequence length is limited to 100 and all the MNMT models are trained for 120K steps on 4 Nvidia RTX A6000 GPUs. We add a target language token l to the source sentence to indicate the language to translate into following

Model	Model Size	en-xx		xx-en		Decoding Speed (tokens/s)
		All	WR	All	WR	
Multilingual	76.96M	24.38	–	31.73	–	1873
+Adapter	224.81M	+2.85	93.62	+1.78	88.30	1726
+CLSR	136.08M	+2.12	94.68	+1.25	88.30	1380
sCLM-top	179.87M	+2.91	**96.81**	+1.62	94.68	1564
mCLM-top	224.61M	+3.13	95.74	+1.83	91.49	1590
sCLM	179.89M	**+3.19**	95.74	+1.91	**97.87**	1143
mCLM	224.63M	+3.01	94.68	**+2.14**	92.55	1240

Table 9: Comparisons of translation quality and decoding speed on the OPUS-100 training data. The bottleneck dimension in Adapter is set to 128. The feature number k is set to 194 in *sCLM* models for both en-xx and xx-en translation, while k is set to 134/154 in *mCLM* models for en-xx and xx-en translation, respectively.

Johnson et al. (2017). However, the language token l is altered to denote the source language in our experiments when performing xx-en translation following Zhang et al. (2021).

We average the last 5 checkpoints for evaluation. We perform beam search decoding with beam size of 4 and length penalty of 1.0.

C More Results

C.1 Comparisons on Performance and Speed

We compare the translation performance and decoding speed of our methods with all the baselines. For fair comparisons, we build another CLM variant (*CLM-top*) in which the CLM modules are only introduced in each feed-forward sublayer similar to Adapter. The results are listed in Table 9. We give two major findings:

- Compared with the original *CLM* models, the *CLM-top* models suffer from slight degradation in most cases, showing that it is better to introduce CLM modules in all the sublayers. Despite that, the *CLM-top* models can achieve similar or better performance compared with Adapter and CLSR. These results further show the effectiveness of our method.

- The decoding speed is related to both the amount of the CLM modules in Transformer and the number of features in each CLM module. Compared with Adapter, all the CLM models slow down the decoding speed due to the token-level feature mixing.

C.2 Detailed Results on Feature Proportion Similarity

We show the top-5 similar languages for each language based on their feature proportion similarity. The results in the encoder and the decoder are listed in Tables 10 and 11, respectively.

C.3 Visualization of Sentence Representations

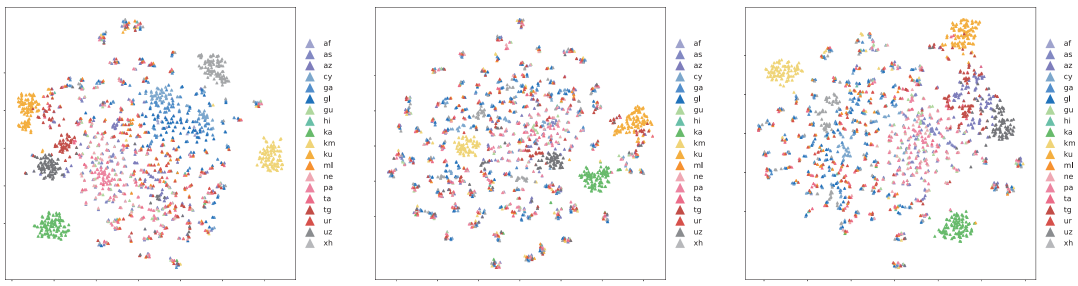
The visualizations on med- and high-resource languages are depicted in Figures 5 and 6, respectively.

Code	Language	Genus	Family	Similar Languages	Code	Language	Genus	Family	Similar Languages
af	Afrikaans	Germanic	Indo-European	fy nl de nn nb	sq	Albanian	Albanian	Indo-European	it es pl ro pt
da	Danish	Germanic	Indo-European	sv nb no nl nn	br	Breton	Celtic	Indo-European	as cy bn pl it
de	German	Germanic	Indo-European	nl ru da fr nb	cy	Welsh	Celtic	Indo-European	fy km nn kk as
fy	Western Frisian	Germanic	Indo-European	af nn pa ne li	ga	Irish	Celtic	Indo-European	fr ru gd sh mt
is	Icelandic	Germanic	Indo-European	no sv da nl bs	gd	Gaelic	Celtic	Indo-European	ga km af or nn
li	Limburgan	Germanic	Indo-European	fy tk yi ku ky	el	Greek	Greek	Indo-European	si cs pl mk sk
nl	Dutch	Germanic	Indo-European	de sv da no ru	ja	Japanese	Japanese	Japanese	ko ml bn si th
no	Norwegian	Germanic	Indo-European	sv da is nb nl	ko	Korean	Korean	Korean	ja ml th si bn
nb	Norwegian Bokmål	Germanic	Indo-European	da nn sv no de	rw	Kinyarwanda	Bantoid	Niger-Congo	be fy oc ne km
nn	Norwegian Nynorsk	Germanic	Indo-European	nb da sv fy no	xh	Xhosa	Bantoid	Niger-Congo	zu et ru es ku
sv	Swedish	Germanic	Indo-European	da no nb is nl	zu	Zulu	Bantoid	Niger-Congo	xh fy kk wa ne
yi	Yiddish	Germanic	indo-European	li fy as ne ky	ig	Igbo	Igboid	Niger-Congo	cy fy li km ky
as	Assamese	Indic	Indo-European	ne or gu pa bn	az	Azerbaijani	Turkic	Altaic	ug tt ur uz am
bn	Bengali	Indic	Indo-European	ml ko hi ja as	kk	Kazakh	Turkic	Altaic	ky be or ne fy
gu	Gujarati	Indic	Indo-European	ne pa or as km	ky	Kyrgyz	Turkic	Altaic	be kk nn fy ne
hi	Hindi	Indic	Indo-European	ur ta ug tg bn	tk	Turkmen	Turkic	Altaic	li ku fy ky ps
mr	Marathi	Indic	Indo-European	or bn hi ml uk	tr	Turkish	Turkic	Altaic	ko ja ml bs pl
ne	Nepali	Indic	Indo-European	gu pa as or fy	tt	Tatar	Turkic	Altaic	az ug uz ur tg
or	Oriya	Indic	Indo-European	pa gu as ne kn	ug	Uyghur	Turkic	Altaic	az ur tt hi uz
pa	Panjabi	Indic	Indo-European	ne gu or as fy	uz	Uzbek	Turkic	Altaic	tt ug az ur tg
si	Sinhala	Indic	Indo-European	ml el ko ja bn	am	Amharic	Semitic	Afro-Asiatic	az tg ur ug hi
ur	Urdu	Indic	Indo-European	hi tg ug az ta	ar	Arabic	Semitic	Afro-Asiatic	af ru es it pt
fa	Persian	Iranian	Indo-European	ko vi uk ml hi	he	Hebrew	Semitic	Afro-Asiatic	hr pl bs uk sr
ku	Kurdish	Iranian	Indo-European	ta hi uz ur tg	mt	Maltese	Semitic	Afro-Asiatic	fr it sh de es
ps	Pashto	Iranian	Indo-European	gu or ne pa as	ha	Hausa	West Chadic	Afro-Asiatic	ur tg az ug hi
tg	Tajik	Iranian	Indo-European	ur hi ug az am	et	Estonian	Finnic	Uralic	fi ru de cs uk
ca	Catalan	Romance	Indo-European	es gl it pt sr	fi	Finnish	Finnic	Uralic	et hu pl cs uk
es	Spanish	Romance	Indo-European	pt gl it ca fr	hu	Hungarian	Ugric	Uralic	fi cs et pl sk
fr	French	Romance	Indo-European	it es pt ru de	km	Central Khmer	Khmer	Austro-Asiatic	gu be nn fy oc
gl	Galician	Romance	Indo-European	pt es ca it ro	vi	Vietnamese	Viet-Muong	Austro-Asiatic	ms id th ko uk
it	Italian	Romance	Indo-European	es pt fr gl ca	mg	Malagasy	Barito	Austronesian	ms id fr ru es
oc	Occitan	Romance	Indo-European	be km fy se pt	id	Indonesian	Malayo-Sumbawan	Austronesian	ms vi th mg uk
pt	Portuguese	Romance	Indo-European	es gl it ca fr	ms	Malay	Malayo-Sumbawan	Austronesian	id vi th mg uk
ro	Romanian	Romance	Indo-European	it es ca gl pt	kn	Kannada	Southern Dravidian	Dravidian	or ne as pa kk
be	Belorusian	Slavic	Indo-European	ky ru kk km oc	ml	Malayalam	Southern Dravidian	Dravidian	si ko ja bn ta
bg	Bulgarian	Slavic	Indo-European	ka mk uk pl bs	ta	Tamil	Southern Dravidian	Dravidian	hi ml ur bn ku
bs	Bosnian	Slavic	Indo-European	hr sr sl pl mk	te	Telugu	Southern-central Dravidian	Dravidian	ta ml or ne as
cs	Czech	Slavic	Indo-European	sk sl pl hr bs	eu	Basque	Basque	Basque	it et es pt ru
hr	Croatian	Slavic	Indo-European	bs sr sl pl cs	my	Burmese	Burmese-Lolo	Sino-Tibetan	kn or ta kk as
mk	Macedonian	Slavic	Indo-European	bg ka bs sr hr	zh	Chinese	Chinese	Sino-Tibetan	lv ru lt fr bn
pl	Polish	Slavic	Indo-European	cs sk uk sl bs	th	Thai	Kam-Tai	Tai-Kadai	vi ko ms ja ml
ru	Russian	Slavic	Indo-European	uk mk sk de bg	lt	Lithuanian	Baltic	Indo-European	lv sh ru fr et
sh	Serbo-Croatian	Slavic	Indo-European	lv ru lt sk sl	lv	Latvian	Baltic	Indo-European	lt sh ru fr et
sk	Slovak	Slavic	Indo-European	cs sl pl hr bs	ka	Georgian	Kartvelian	Kartvelian	bg mk uk bs sr
sl	Slovenian	Slavic	Indo-European	sk cs hr bs sr	eo	Esperanto	-	-	it uk es ca pl
sr	Serbian	Slavic	Indo-European	bs hr sl mk pl	se	Northern Sami	-	-	fy km pa oc be
uk	Ukrainian	Slavic	Indo-European	pl ru mk bs bg	wa	Wallon	-	-	ne oc fy km pa

Table 10: Top-5 languages similar to anchor language according to the cosine similarity of feature proportions in the *sCLM* encoder on en-xx translation. The languages are categorized based on the typological knowledge base WALS (Dryer and Haspelmath, 2013).

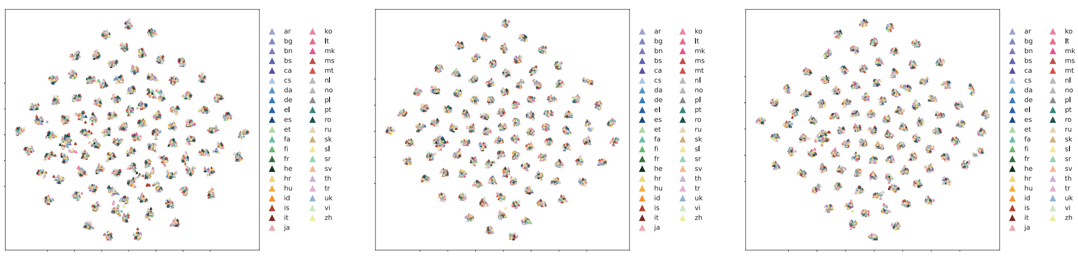
Code	Language	Genus	Family	Similar Languages	Code	Language	Genus	Family	Similar Languages
af	Afrikaans	Germanic	Indo-European	fy li nl nn nb	sq	Albanian	Albanian	Indo-European	ro et sl cs sk
da	Danish	Germanic	Indo-European	no sv nb nn is	br	Breton	Celtic	Indo-European	cy oc ku se wa
de	German	Germanic	Indo-European	nl da nb no sv	cy	Welsh	Celtic	Indo-European	br oc se ku af
fy	Western Frisian	Germanic	Indo-European	af li nn oc nb	ga	Irish	Celtic	Indo-European	gd de nb oc se
is	Icelandic	Germanic	Indo-European	sv no da nb et	gd	Gaelic	Celtic	Indo-European	ga oc cy se ig
li	Limburgan	Germanic	Indo-European	fy af wa nn oc	el	Greek	Greek	Indo-European	ro ka he th no
nl	Dutch	Germanic	Indo-European	af de da no sv	ja	Japanese	Japanese	Japanese	zh ko ta th si
no	Norwegian	Germanic	Indo-European	da sv nb nn is	ko	Korean	Korean	Korean	th ja si zh ta
nb	Norwegian Bokmål	Germanic	Indo-European	nn da no sv af	rw	Kinyarwanda	Bantoid	Niger-Congo	tk li fy af zu
nn	Norwegian Nynorsk	Germanic	Indo-European	nb no da af sv	xh	Xhosa	Bantoid	Niger-Congo	zh sh tk et mt
sv	Swedish	Germanic	Indo-European	da no nb is nn	zu	Zulu	Bantoid	Niger-Congo	xh tk ig ku oc
yi	Yiddish	Germanic	indo-European	gu li af ky kn	ig	Igbo	Igboid	Niger-Congo	zu tk gd rw li
as	Assamese	Indic	Indo-European	bn gu hi he nn	az	Azerbaijani	Turkic	Altaic	tr uz tk gu et
bn	Bengali	Indic	Indo-European	as he gu si hi	kk	Kazakh	Turkic	Altaic	ky be ru uk fy
gu	Gujarati	Indic	Indo-European	ne as hi bn pa	ky	Kyrgyz	Turkic	Altaic	kk be ru uk uz
hi	Hindi	Indic	Indo-European	ne mr gu cs si	tk	Turkmen	Turkic	Altaic	ku oc tr zu cy
mr	Marathi	Indic	Indo-European	hi ne cs gu sk	tr	Turkish	Turkic	Altaic	az tk eo et uz
ne	Nepali	Indic	Indo-European	hi gu mr nn pa	tt	Tatar	Turkic	Altaic	uz kk ky tg he
or	Oriya	Indic	Indo-European	hi gu kn ko pa	ug	Uyghur	Turkic	Altaic	ps ur hi uz ky
pa	Panjabi	Indic	Indo-European	km gu ne ko ja	uz	Uzbek	Turkic	Altaic	tg ky az tt tk
si	Sinhala	Indic	Indo-European	ml ko he th hi	am	Amharic	Semitic	Afro-Asiatic	ky gu uz az or
ur	Urdu	Indic	Indo-European	fa he hi th ar	ar	Arabic	Semitic	Afro-Asiatic	fa he ur de th
fa	Persian	Iranian	Indo-European	ar ur th he de	he	Hebrew	Semitic	Afro-Asiatic	bn si ka ro bg
ku	Kurdish	Iranian	Indo-European	cy br oc se tk	mt	Maltese	Semitic	Afro-Asiatic	it fr sh wa lv
ps	Pashto	Iranian	Indo-European	nn gu ug zu oc	ha	Hausa	West Chadic	Afro-Asiatic	tg ig ku ms tk
tg	Tajik	Iranian	Indo-European	uz be ky ru uk	et	Estonian	Finnic	Uralic	fi ms id ro sq
ca	Catalan	Romance	Indo-European	es gl pt it fr	fi	Finnish	Finnic	Uralic	et eu no id hu
es	Spanish	Romance	Indo-European	gl pt ca it fr	hu	Hungarian	Ugric	Uralic	et fi eo cs es
fr	French	Romance	Indo-European	ca pt es it gl	km	Central Khmer	Khmer	Austro-Asiatic	pa se gu oc nb
gl	Galician	Romance	Indo-European	pt es ca it fr	vi	Vietnamese	Viet-Muong	Austro-Asiatic	ms id et ka no
it	Italian	Romance	Indo-European	pt ca es gl fr	mg	Malagasy	Barito	Austronesian	sh fr fi de lv
oc	Occitan	Romance	Indo-European	wa se cy gl ca	id	Indonesian	Malayo-Sumbawan	Austronesian	ms vi et he fi
pt	Portuguese	Romance	Indo-European	gl es ca it fr	ms	Malay	Malayo-Sumbawan	Austronesian	id vi et ka fi
ro	Romanian	Romance	Indo-European	ca pt gl es it	kn	Kannada	Southern Dravidian	Dravidian	te or km nn ne
be	Belorusian	Slavic	Indo-European	uk ru ky kk tg	ml	Malayalam	Southern Dravidian	Dravidian	si ko vi ta hi
bg	Bulgarian	Slavic	Indo-European	mk ru uk ka he	ta	Tamil	Southern Dravidian	Dravidian	ko gu ja ml hi
bs	Bosnian	Slavic	Indo-European	hr sr sl sk sh	te	Telugu	Southern-central Dravidian	Dravidian	kn vi hi ml ko
cs	Czech	Slavic	Indo-European	sk sl pl hr bs	eu	Basque	Basque	Basque	fi eo id ms gl
hr	Croatian	Slavic	Indo-European	bs sr sl sk sh	my	Burmese	Burmese-Lolo	Sino-Tibetan	gu or eo oc tk
mk	Macedonian	Slavic	Indo-European	bg ru uk ka he	zh	Chinese	Chinese	Sino-Tibetan	ja th ko bn ta
pl	Polish	Slavic	Indo-European	sk cs hr sl bs	th	Thai	Kam-Tai	Tai-Kadai	ko zh si ru uk
ru	Russian	Slavic	Indo-European	uk bg be mk ky	lt	Lithuanian	Baltic	Indo-European	lv sh eo ru cs
sh	Serbo-Croatian	Slavic	Indo-European	hr sr bs sl sk	lv	Latvian	Baltic	Indo-European	lt sh et nb ru
sk	Slovak	Slavic	Indo-European	cs sl pl hr bs	ka	Georgian	Kartvelian	Kartvelian	bbg mk ru he nl
sl	Slovenian	Slavic	Indo-European	hr bs sr sk cs	eo	Esperanto	-	-	ca es gl oc pt
sr	Serbian	Slavic	Indo-European	bs hr sl sk sh	se	Northern Sami	-	-	oc cy ku nn br
uk	Ukrainian	Slavic	Indo-European	ru bg be mk th	wa	Wallon	-	-	oc af ku nn li

Table 11: Top-5 languages similar to anchor language according to the cosine similarity of feature proportions in the *sCLM* decoder on en-xx translation.



(a) Multilingual baseline (b) *sCLM* (c) *mCLM*

Figure 5: t-SNE visualizations of the encoder representations of 19 med-resource languages on xx-en translation encoded by Multilingual baseline, *sCLM* and *mCLM*.



(a) Multilingual baseline (b) *sCLM* (c) *mCLM*

Figure 6: t-SNE visualizations of the encoder representations of 41 high-resource languages on xx-en translation encoded by Multilingual baseline, *sCLM* and *mCLM*.